

AD-A062 389

NOTRE DAME UNIV IND DEPT OF AEROSPACE AND MECHANICA--ETC F/G 12/2
OPTIMIZATION OF MULTIPLE-RESPONSE SIMULATION MODELS.(U)
OCT 78 W E BILES

N00014-76-C-1021
NL

UNCLASSIFIED

1 OF 2
ADA
062389

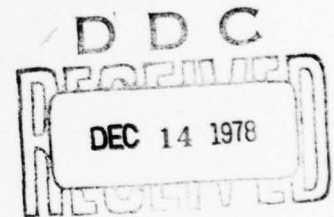
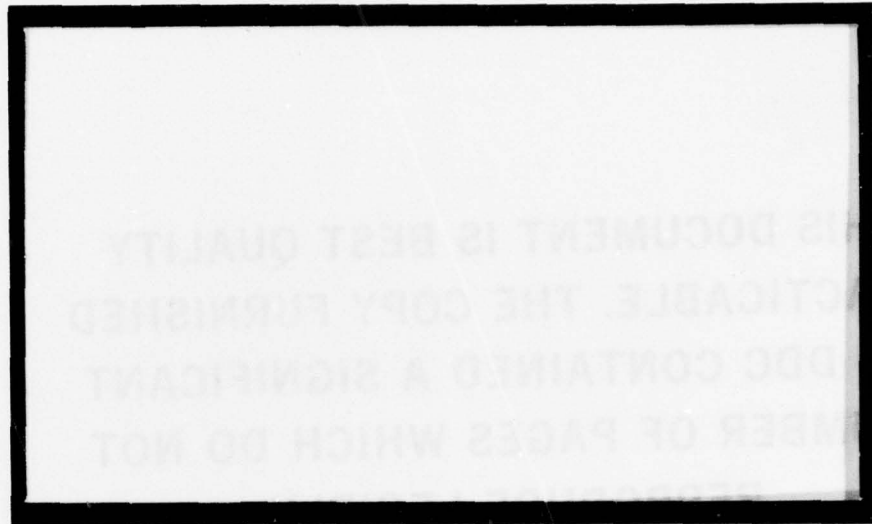




DDC FILE COPY AD A062389

LEVEL

12



CF F

INDUSTRIAL ENGINEERING



UNIVERSITY OF NOTRE DAME, NOTRE DAME, INDIANA 46556

Department of Aerospace and Mechanical Engineering

78 12 11 160

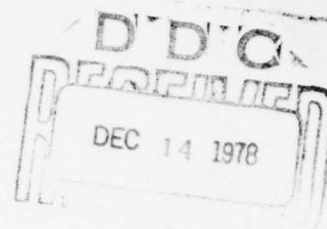
DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

AD A062389

DDC FILE COPY

9 FINAL REPORT
1 Sep 76-31 Oct 78



6 OPTIMIZATION OF MULTIPLE-RESPONSE
SIMULATION MODELS

10 William E. Biles

Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

This document has been approved
for public release and sale; its
distribution is unlimited.

11 31 Oct 78

October 31, 1978

12 96p.

This research was sponsored by
Naval Analysis Programs
Office of Naval Research
Contract N00014-76-C-1021

15 NR 277-237

16 RR01411

17 RR0141101

Reproduction in whole or in part is permitted for any purpose of the United States Government. Available for public release; distribution unlimited.

401 829

JOB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) OPTIMIZATION OF MULTIPLE RESPONSE SIMULATION MODELS.		5. TYPE OF REPORT & PERIOD COVERED Final Report September 1, 1976-Oct. 31, 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dr. William E. Biles		8. CONTRACT OR GRANT NUMBER(s) N0014-76-C-1021e
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Aerospace and Mechanical Engr. University of Notre Dame Notre Dame, IN 46556		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153 N RR014-11-01 NR 277-237
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Analysis Programs (Code 431) Office of Naval Research Arlington, VA 22217		12. REPORT DATE October 31, 1978
		13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) N/A		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A
16. DISTRIBUTION STATEMENT (of this Report) Available for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Available for public release; distribution unlimited.		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Optimization Simulation Response Surface Methodology		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes several computerized multiple-variable, multiple-response optimization procedures for use in connection with computer simulation models. These procedures include (1) "complex" search, (2) a sequential first-order response surface approach, (3) a second-order response surface method, and (4) "complex" search followed by second-order response surface analysis. These four optimization procedures are evaluated with (1) a stochastic inventory simulation model, (2) a tank duel simulation, and (3) a naval minefield evaluation model.		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

78-12-11-100
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Page

ABSTRACT	ii
INTRODUCTION	1
BACKGROUND	2
PROBLEM FORMULATIONS	5
Constrained Optimization	7
Multiple-Objective Optimization	7
OPTIMIZATION TECHNIQUES	10
Direct Search Methods	11
First-Order Response Surface Methods	15
Second-Order Response Surface Methods	23
Second-Order Response Surface Analysis of Box's Complex Search Experiments	32
SIMULATION TEST MODELS	36
Stochastic Inventory System	36
Tank Duel Model	37
Mine Hunting	39
Air Attack on a Surface Fleet	42
DISCUSSION OF RESULTS	45
Stochastic Inventory Model	45
Tank Duel Model	47
Mine Hunting Model	47
SUMMARY AND CONCLUSIONS	50
REFERENCES	54
APPENDICES	
Appendix A - Program Listing for an Experimental Box Complex Search	56
Appendix B - Program Listing for a Sequential First- Order Response Surface Method	61
Appendix C - Program Listing for a Second-Order Response Surface Method Based on a Central Composite Design	70
Appendix D - Program Listing for a Second-Order Response Surface Method Based on Box's Complex Search	79
DISTRIBUTION	90

ABSTRACT

This report describes several computerized multiple-variable, multiple-response optimization procedures developed for use in connection with simulation models. The four optimization procedures evaluated in this research were as follows:

1. Box's complex search method [8];
2. A sequential first-order response surface approach [2,3];
3. A second-order response-surface experimental design followed by mathematical optimization;
4. A Box complex search followed by a second-order response surface optimization (methods 1 and 3 combined).

Each of these optimization methods involved an adaptation of the original procedure on which it was based, in order to accommodate multiple simulation responses η_j , $j=1, \dots, m$.

These four optimization procedures were evaluated with three computer simulation models:

- i. A stochastic inventory model [15,16];
- ii. A tank duel model [21];
- iii. A minefield evaluation model [1].

Exploratory work was also performed with a simulation model of a missile attack on a surface fleet [18].

This research has produced workable optimization procedures for interfacing with simulation models of naval operations, but it is concluded that this interface must be managed on a "customized" basis for each simulation model due to the variety of approaches employed in naval systems modeling.

INTRODUCTION

The design and analysis of complex systems, especially those involving probabilistic or stochastic elements, often necessitates the use of digital computer simulation. A problem-solving procedure for defining and analyzing a model of a system, simulation affords the analyst the opportunity to evaluate complex systems, (a) without constructing them if they are proposed systems, (b) without interrupting their operation if they are actual systems, or (c) without damaging or destroying them if the purpose of simulation is to test the effectiveness of the system in a hazardous operating environment.

Naval operations present especially challenging problems for the simulationist. The purpose in simulating naval operations would very likely be to evaluate the effectiveness of certain offensive or defensive fleet configurations and tactics. Large numbers of controllable input variables, uncontrollable input variables, and measures of system effectiveness are often involved in realistic models of naval operations. This not only complicates model development, but hampers experimentation with the model as well.

Simulation can be defined as the establishment of a mathematical-logical model of a system and the experimental manipulation of that model on a digital computer. This paper concentrates on the latter aspect, particularly upon means of determining an "optimal" design through experimentation with a digital computer simulation model. (The word "optimal" is used guardedly here, since the randomness inherent in computer simulation contradicts the classical notion of an optimal solution.) This research has explored procedures for combining computer simulation, response surface methodology and mathematical programming to seek an estimated "optimal"

solution. These techniques are demonstrated and compared using simulation models of (1) a stochastic inventory system (2) a task duel, and (3) mine-field evaluation.

BACKGROUND

A computer simulation model can be regarded as a "black-box", as illustrated in Figure 1, in which values for n controllable input variables x_i , $i=1, \dots, n$ are combined in some manner to produce values for a set of m output or response variables η_j , $j=1, \dots, m$. These responses are measures of merit or effectiveness for the system being modeled. It is typically true that the system responses are affected by a set of p uncontrollable factors, z_k , $k=1, \dots, p$.

The execution of the computer simulation model, either for some fixed simulated time t or until r_j realizations of the j -th system response have been obtained, produces m simulated responses.

$$y_j = g_j(x_1, \dots, x_n | z_1, \dots, z_p) + \epsilon_j, \quad j=1, \dots, m \quad (1)$$

The quantity y_j is an estimate of the true system response η_j , and is found from the relation

$$y_j = \frac{1}{r_j} \sum_{\ell=1}^{r_j} \xi_{j\ell}, \quad j=1, \dots, m \quad (2)$$

where $\xi_{j\ell}$ is the individual value of the j -th system response at the ℓ -th realization and r_j is the number of realizations of this response. Equation (2) suggests that, for a given execution, or simulation trial, the r_j , $j=1, \dots, m$ need not be the same. The simulation trial also produces an unbiased estimate of the variance of the j -th system response; that is,

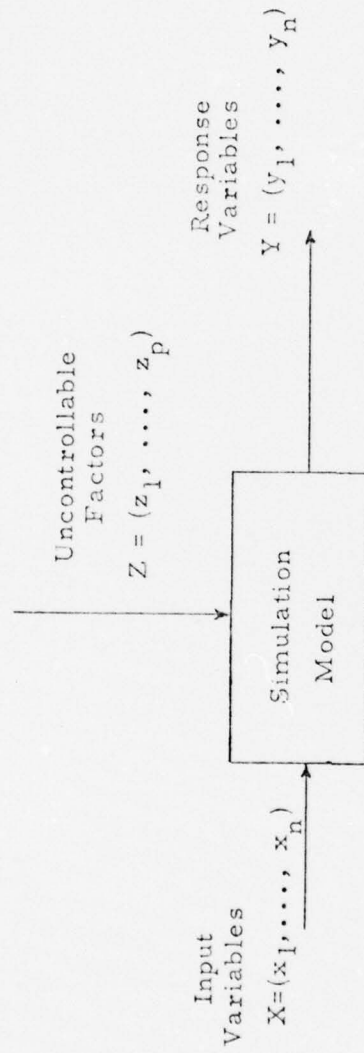


Figure 1. A "black-box" view of computer simulation

$$s_j^2 = \frac{1}{(r_j - 1)} \sum_{\ell=1}^{r_j} (\xi_{j\ell} - y_j)^2 \quad (3)$$

or more conveniently

$$s_j^2 = \frac{1}{(r_j - 1)} \left[\sum_{\ell=1}^{r_j} \xi_{j\ell}^2 - r_j y_j^2 \right] \quad (3a)$$

Fishman [12] and Kleijnen [19] describe other procedures for computing s_j^2 in the case of serial correlation of the relationships $\xi_{j\ell}$.

The structure of equation (1) must be carefully examined in order to develop the basic framework for the procedures presented here. It is stated that y_j is an estimate of the true response η_j ; hence,

$$y_j = \eta_j + \epsilon_j \quad (4)$$

The ϵ_j term reflects the random variation inherent in the probabilistic system under study, and can be regarded as sampling error. The uncontrollable factors, the $z_k, k=1, \dots, p$ terms in equation (1), can be viewed as contributing to the random variation ϵ_j . Thus equation (1) can be simplified to

$$y_j = g_j(x_1, \dots, x_n) + \epsilon_j, \quad j=1, \dots, m \quad (5)$$

The estimate y_j is therefore a random variable with mean

$$E(y_j) = \eta_j = g_j(x_1, \dots, x_n), \quad j=1, \dots, m \quad (6)$$

and variance

$$\text{Var}(y_j) = \text{Var}(\epsilon_j), \quad j=1, \dots, m \quad (7)$$

This statement suggests that the true response η_j is a point on an unknown $(n+1)$ -dimensional hypersurface $g_j(X)$ at a given point $X = (x_1, \dots, x_n)$ and that the simulation-generated response y_j is a variate from a probability

distribution about the true value η_j at this point X.

The response surface methods employed in this methodology require the assumption and the random error ϵ_j be normally distributed with mean zero and variance σ_j^2 ; that is,

$$E(y_j | x_1, \dots, x_n) = \eta_j \quad j=1, \dots, m \quad (8)$$

$$\text{Var}(y_j | x_1, \dots, x_n) = \sigma_j^2, \quad j=1, \dots, m \quad (9)$$

It is also required that $\text{Var}(y_j | x_1, \dots, x_n)$ be the same at any point on the $(n+1)$ -dimensional hypersurface.

Note that there are m distinct $(n+1)$ -dimensional hypersurfaces, each having its own characteristic random variation. The problem at hand is to have the computer simulation model which produces these m responses be either automatically or interactively controlled by an "optimization" procedure, as illustrated in Figure 2. In general, the "optimization" procedure must combine the experimental features of response surface methodology with the logical/numerical procedures of mathematical programming.

PROBLEM FORMULATIONS

We shall consider two basic approaches to formulating the problem of optimizing a multiple-variable, multiple-response simulation model. One approach is the familiar constrained optimization formulation in which one of the simulation responses, say η_1 , is to be maximized or minimized, subject to maintaining the remaining $m-1$ responses within prescribed bounds. The second approach is the multiple-objective formulation, in which the m responses are either weighted to form a single objective or treated in a manner akin to goal programming. Each of these two formulations is described in the following sections.

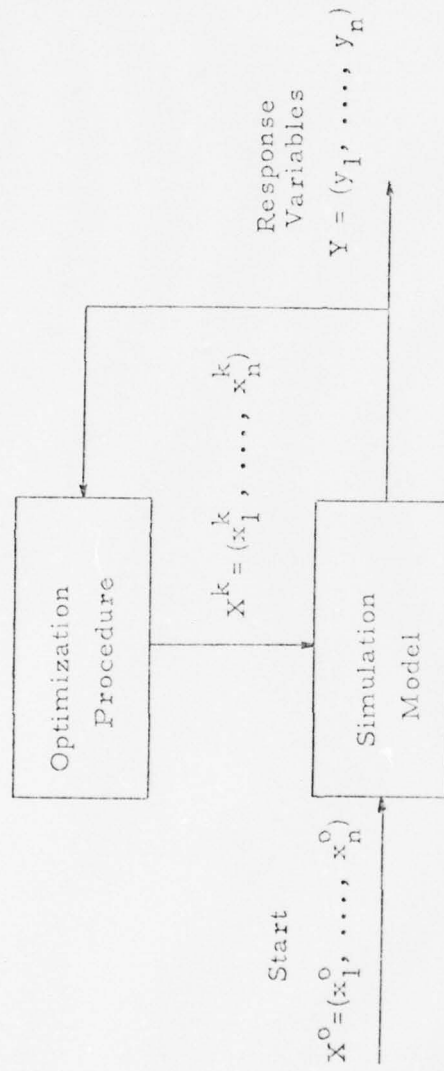


Figure 2. The interface between optimization and simulation

Constrained Optimization

Under the constrained optimization approach, the problem is stated as

$$\text{Max(or Min)} \quad \eta_1 = g_1(x_1, \dots, x_n) \quad (10)$$

subject to the constraints

$$a_i \leq x_i \leq c_i, \quad i=1, \dots, n \quad (11)$$

$$\eta_j = g_j(x_1, \dots, x_n) \quad \left\{ \begin{array}{l} > \\ = \\ < \end{array} \right\} d_j, \quad j=2, \dots, m \quad (12)$$

The constraints expressed in (11) are bounds on the controllable input variables x_1, \dots, x_n and are typically known a priori. The bounds (11) generally form the known experimental region prior to conducting simulation trials. In contrast to that, the response functions $g_j(x_1, \dots, x_n)$ in (12) are not usually known a priori and hence the responses η_j must be estimated experimentally via simulation. Thus, simulation trials performed at points satisfying (11) may yield responses violating (12).

To complicate matters even more, the random error ϵ_j can lead to erroneous decisions relative to feasibility with respect to the constraints in (12). The same is true relative to the objective function in (10). That is, one simulation trial can appear to represent an improvement over another when the true response at this particular set of values x_1, \dots, x_n does not. These difficulties can be countered through variance reduction techniques, as discussed by Fishman [12] and Kleijnen [19].

Multiple-Objective Optimization

One approach to a multiple-objective formulation is to assign weights $w_j, j=1, \dots, m$ to the m responses and form a single objective function

$$\text{Max (or Min) } W = \sum_{j=1}^m w_j g_j(x_1, \dots, x_n) \quad (13)$$

The bounds (11) still apply, so that the problem remains one of constrained optimization, but one in which the entire feasible region is known a priori. The weight $w_j, j=1, \dots, m$ are typically assigned through the subjective judgment of the decision-maker in the system being simulated. These weights are usually normalized, so that

$$\sum_{j=1}^m w_j = 1 \quad (14)$$

One frequently encounters the situation in which certain of the responses η_j are to be maximized and other minimized. This case is handled by maximizing the negative of those functions which are to be minimized, so that the objective function in (13) is rearranged to the form

$$\text{Max } W = \sum_{j=1}^s w_j g_j(x_1, \dots, x_n) - \sum_{j=s+1}^m w_j g_j(x_1, \dots, x_n) \quad (15)$$

where s functions are maximized and $m-s$ functions are minimized.

A second approach to the multiple-objective formulation is one which casts the objective function as a "utility function"; that is,

$$\text{Max } U[g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)] \quad (16)$$

subject to the bounds in (11). The formulation in (15) is a special case of that in (16), in which $U[g_j(x_1, \dots, x_n)]$ is a linear additive function. Montgomery and Bettencourt [21] discuss various formulations of the multiple objective optimization problem, as well as several approaches to its solution, and demonstrate its application to multiple-response simulation.

Another multiple-objective optimization formulation is that called

goal programming. This procedure is initiated by establishing a set of goals in terms of the \underline{m} system responses. These goals are expressed as

$$G_j = g_j(x_1, \dots, x_n), \quad j=1, \dots, m \quad (17)$$

Each goal must have an associated right-side value d_j ; that is,

$$G_j = g_j(x_1, \dots, x_n) \begin{cases} \leq \\ = \\ \geq \end{cases} d_j, \quad j=1, \dots, m \quad (18)$$

With a slight modification, each goal can be expressed as an equality

$$G_j = g_j(x_1, \dots, x_n) + n_j - p_j = d_j, \quad j=1, \dots, m \quad (19)$$

where n_j is a negative deviation from d_j , and p_j is a positive deviation. Either n_j or p_j must be zero in any given solution, and both could be zero. Next, each of the \underline{m} goals G_j is assigned to a priority level P_k , $k=1, \dots, \ell$, where P_1 represents the highest priority and P_ℓ the lowest. For any goal falling within a given priority level P_k , the decision-maker can weigh one goal relative to another. The final step in problem formulation is to combine these several levels of goals into an achievement function which has the form

$$A = \{P_1(\bar{n}_1, \bar{p}_1), P_2(\bar{n}_2, \bar{p}_2), \dots, P_\ell(\bar{n}_\ell, \bar{p}_\ell)\} \quad (20)$$

This achievement function is simply an ordered ℓ -vector. Its structure is predicated on one of the following procedures for achieving the j -th goal:

- (a) To equal or exceed d_j , minimize n_j
- (b) To equal or be less than d_j , minimize p_j
- (c) To equal d_j , minimize $(n_j + p_j)$

A solution (x_1^*, \dots, x_n^*) is considered optimal if the corresponding value A^* is the same as or preferred to any other value A . Therefore, the general

goal programming problem is to find x_1, \dots, x_n so as to minimize the ordered vector (20) such that the goals (19) are satisfied and

$$\begin{aligned} x_i &\geq 0, \quad i=1, \dots, n \\ n_j &\geq 0, \quad j=1, \dots, m \\ p_j &\geq 0, \quad j=1, \dots, m \end{aligned} \tag{21}$$

The functions $g_j(x_1, \dots, x_n)$ in (19) are generally unknown, but are usually assumed to be nonlinear. Any technique proposed for solving this problem in the simulation domain must provide experimental estimates of these unknown functions, as well as a mathematical procedure for optimization. Moreover, the experimental observations are produced via simulation - each simulation trial at a point x_1, \dots, x_n produces m responses $y_j=1, \dots, m$. Biles [4] has described the application of nonlinear goal programming to the multiple-response simulation problem, based on techniques proposed by Ignizio [17].

OPTIMIZATION TECHNIQUES

Various procedures have been applied in combining optimization and simulation to seek the "optimum" solution to systems possessing a single response η . The multiple-response problem described here is complicated by the necessity to observe several responses at once, and to incorporate these values into the optimization technique. But many of the same techniques that have been applied successfully to the single-response problem can, with appropriate modifications, be extended to accommodate multiple responses. Moreover, these modified procedures are often applicable to more than one of the aforementioned formulations of the multiple-response problem.

The optimization procedures described below fall into three categories: (1) direct search techniques, (2) first-order response surface methods, and

(3) second-order response surface procedures. Although numerous techniques will be cited, only a few broadly stated procedures will be outlined here. It should be remembered that, although we may refer to "optimization" techniques, the classical notion of an "optimum" solution is inapplicable due to the presence of the sampling error ϵ_j associated with each response variable η_j . Rather we shall seek a solution which hopefully lies close to the true solution. In a more formal sense, we might state that we are to some degree confident, say 90%, that the true solution lies within some interval about our estimated solution.

Direct Search Methods

Direct search methods are those which, applied in a purely computational manner, do not require the use of derivatives. These methods progress through a sequence of points according to some algorithm. Typical of this class of optimization techniques are the pattern search algorithm by Hooke and Jeeves [14], sequential simplex search by Spendley, Hext and Himsworth [26], and the so-called "complex" search method by M. J. Box [8]. In general, these direct search procedures make rapid early progress toward an "optimum", but iterate laboriously as a solution is neared. This is particularly true in the presence of random error, as encountered in simulation.

Among the direct search techniques, Box's "complex" method [8] has been found to be the easily adapted to a multiple-response environment. It also performs better than any of the other direct search techniques in the face of random error and constraints. In fact, "complex" search is not at all complex, but derives its name from a contraction of the words "constrained simplex": it evolved from the sequential simplex method [26] and the necessity to deal with constraints. This report describes a modification of Box's method which makes it especially suitable for the multiple-response

simulation problem. The following procedure describes a generalized "complex" procedure as it might be applied to the multiple-response simulation problem:

1. Randomly generate a set of $N \geq n + 2$ search points X^1, \dots, X^N satisfying the known bounds (11).
2. Perform a simulation trial at each of these N search points and record the mN estimated responses $y_j^\ell, j=1, \dots, m, \ell=1, \dots, N$.
3. Where a given point X^k is observed to violate one or more constraints, if such constraints apply with the particular problem formulation being employed, generate a replacement search point $X^{k'}$, perform a simulation trial at $X^{k'}$, and record the \underline{m} estimated response at $X^{k'}$.
4. After N feasible search points have been established, evaluate the objective function for each of these N points. This "objective function" might be η_1 in (10), W in (15), U in (16), or A in (20). Among these N search points, find the worst point X^W ; that is, the search point giving the least desirable value of the objective function. Define X^C as the centroid of the $N-1$ remaining points. Project from X^W through X^C to the image point $X^{W'}$. If the known bounds (11) are violated by this move, shorten the step to $X^{W'}$ until no violation occurs. Perform a simulation trial at $X^{W'}$.
5. Repeat steps 3 and 4 until a solution (X^*, Y^*) is obtained which represents the best solution that can be achieved within the available computer time. Figure 3 illustrates Box's complex search as applied to a constrained problem. Figure 4 gives a

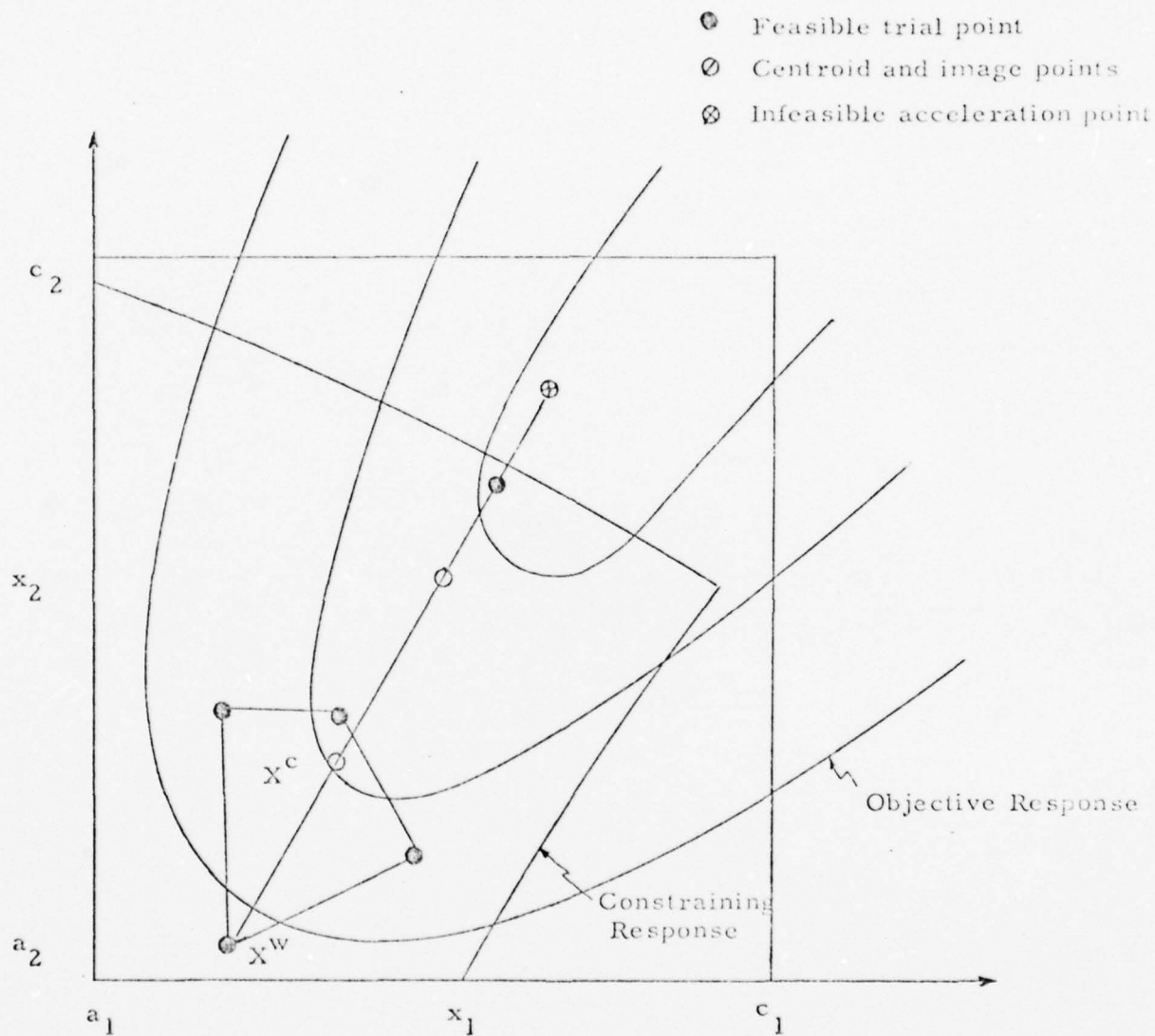


Figure 3. Complex Search

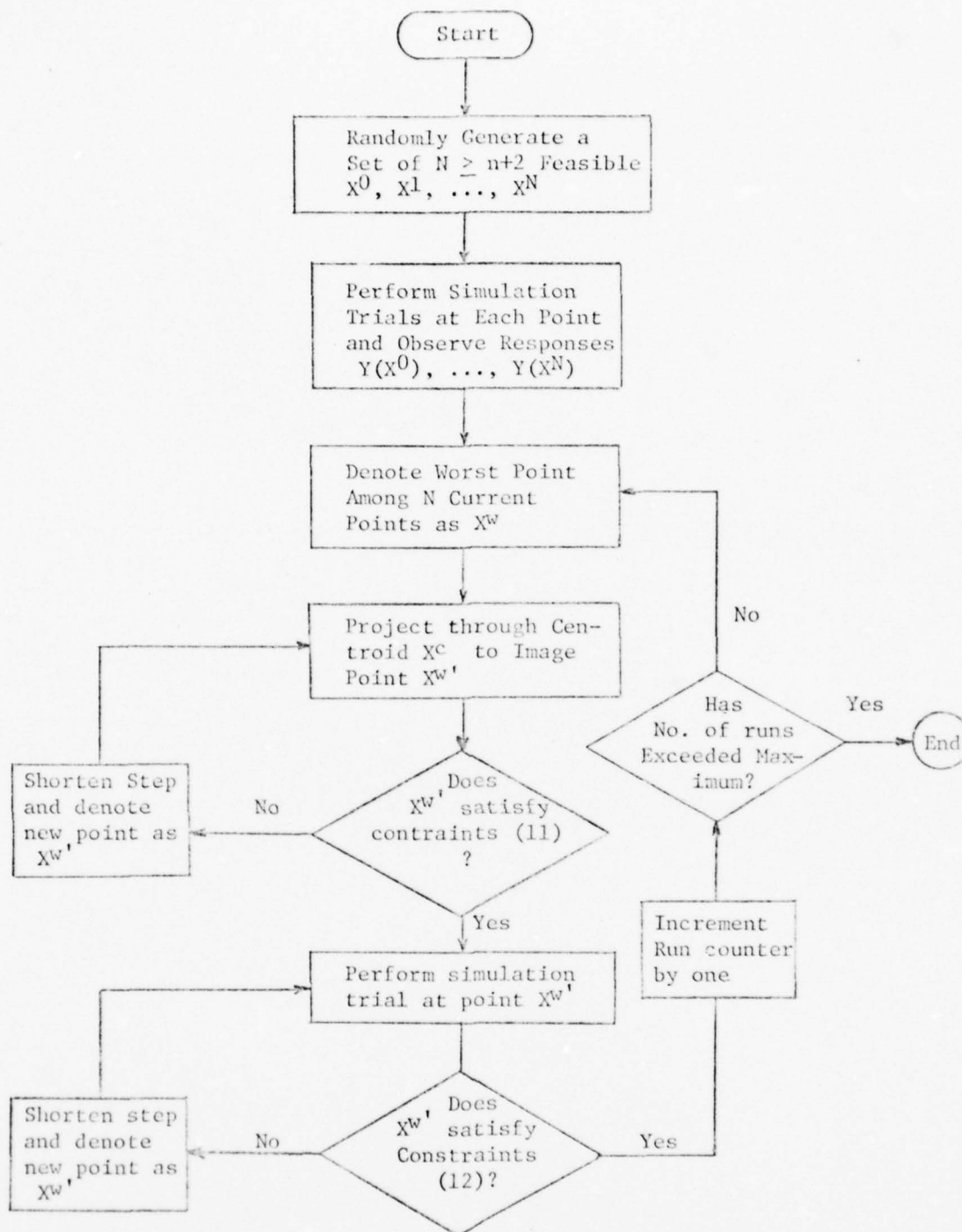


Figure 4. A multiple-response complex search procedure

a flow chart for a simulation/optimization method based on the procedure outlined above. Appendix A contains the program listing for a constrained optimization formulation of the Box complex search method.

A significant advantage of complex search is that, once an initial "complex" of N feasible simulation trials (and perhaps several infeasible trials) have been performed, trials are conducted one at a time thereafter. The search can be continued as long as improved solutions are obtained. If several successive simulations are performed at scattered points around the known experimental region without achieving an improved solution, however, the search can be terminated and the best available solution adopted.

First-Order Response Surface Methods

First-order response surface methods attempt to accomplish experimentally what the "method of steepest ascent" accomplishes computationally. From a current point X^k , a designed experiment is conducted (with a simulation trial at each design point) to estimate the gradient direction $\nabla g(X^k)$. Simulation trials are then conducted at points along this direction to a new point X^{k+1} which represents the best solution obtained along the direction $\nabla g(X^k)$. This process is an experimental approximation of

$$X^{k+1} = X^k + \lambda^k [\nabla g(X^k)] \quad (22)$$

The step length λ^k can be estimated by a line search or by a regression procedure as described by Biles [2,3].

The gradient direction $\nabla g(X^k)$ is estimated by placing an appropriate first-order experimental design, such as a 2^n factorial, 2^{n-p} fractional factorial, or n -dimensional simplex design (see Myers [22]) around the current point X^k . A simulation trial is performed at each point in the

selected experimental design. From these N observations the multiple linear regression model

$$\hat{y} = b_0 + \sum_{i=1}^n b_i x_i \quad (23)$$

can be estimated (see Draper and Smith 10]). Since the gradient direction $\nabla g(X^k)$ is mathematically defined as the n -vector of first partial derivatives of $g(X)$ evaluated at X^k , it is clear that $\nabla g(X^k)$ is simply the n -vector of regression coefficients exclusive of the b_0 term; that is,

$$\nabla g(X^k) = (b_1, \dots, b_n)' \quad (24)$$

In the multiple-response simulation problem, a simulation trial is conducted at each design point in the selected first-order design and the \underline{m} observations y_j^i , $j=1, \dots, m$ are recorded at each design point. Multiple linear regression is applied separately to each set of observations (assuming independence among the \underline{m} responses), producing the \underline{m} models

$$\hat{y}_j = b_{j,0} + \sum_{i=1}^n b_{j,i} x_i, \quad j=1, \dots, m \quad (25)$$

and hence the \underline{m} gradient vectors

$$\nabla g_j(X^k) = (b_{j,1}, \dots, b_{j,n})', \quad j=1, \dots, m \quad (26)$$

These estimates can then be employed in any one of several optimization schemes to produce an improved solution X^{k+1} . A generalized procedure for accomplishing this improved solution, and an estimated "optimum", will be described later. But first it is necessary to give attention to the experimental designs employed to estimate the gradient vectors $\nabla g_j(X^k)$, $j=1, \dots, m$.

In selecting a first-order response surface design, it is usually desirable to minimize the variances of the regression coefficients b_i , $i=1, \dots, n$. To accomplish this the first-order experimental design should be orthogonal.

That is, the placement of the N experimental points (in our case, simulation trials) is described by the N by n design matrix D , where

$$D = \begin{vmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ x_{1N} & x_{2N} & \cdots & x_{nN} \end{vmatrix} \quad (27)$$

Then an N by $n+1$ matrix X is constructed by placing a unit vector to the left of D . Thus,

$$X = \begin{vmatrix} 1 & x_{11} & x_{21} & \cdots & x_{n1} \\ 1 & x_{12} & x_{22} & \cdots & x_{n2} \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 1 & x_{1N} & x_{2N} & \cdots & x_{nM} \end{vmatrix} \quad (28)$$

It is usually convenient to code the design levels, so that the following conditions are achieved:

$$\begin{aligned} \sum_{u=1}^N x_{iu}^2 &= N \\ \sum_{u=1}^N x_{iu} &= 0 \end{aligned} \quad i=1, \dots, n \quad (29)$$

If the actual value of the u -th level of the i -th variable is z_{iu} , then the corresponding coded values is

$$x_{iu} = \frac{z_{iu} - z_{iu}}{S_i} \quad (30)$$

where

$$\bar{z}_i = \left(\sum_{u=1}^N z_{iu} \right) / N \quad (31)$$

and

$$S_i = \sum_{u=1}^N (z_{iu} - \bar{z}_i)^2 / N \quad (32)$$

Then

$$X'X = \begin{vmatrix} N & 0 & 0 & \dots & 0 \\ 0 & N & 0 & \dots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & 0 & 0 & \dots & 0 \end{vmatrix} \quad (33)$$

Since the $(n+1)$ - vector of regression coefficients \bar{b} is estimated by the least squares relation

$$\bar{b} = (X'X)^{-1} X' \bar{y} \quad (34)$$

where \bar{y} is the N -vector of response estimates obtained from N simulation trials, the variance of the regression coefficients b_i , $i=1, \dots, n$ is given by

$$\text{Var}(b_i) = \sigma^2 / N, \quad i=1, \dots, n \quad (35)$$

where σ^2 is the variance of the error term ϵ . Since we are interested in m separate system response y_j , $j=1, \dots, m$, equations (34) and (35) can be generalized to

$$\bar{b}_j = (X'X)^{-1} X' \bar{y}_j, \quad j=1, \dots, m \quad (36)$$

$$\text{Var}(b_{ji}) = \sigma_j^2 / N, \quad \begin{matrix} i=1, \dots, n \\ j=1, \dots, m \end{matrix} \quad (37)$$

For an orthogonal first-order design, the results in (33)-(37) hold, giving a so-called "minimum-variance" design. The 2^n factorial and 2^{n-p} fractional factorial designs are orthogonal and hence minimum variance. Orthogonal n -simplex designs can be easily constructed (see Myers [22]). Since n -simplex designs provide the minimum number of design points needed to estimate the multiple-linear regression models in (23) or (25), and are hence the most "economical" of the first-order response surface designs, they are especially attractive for the purpose proposed here. Brooks and Mickey [9] concluded that n -simplex designs offer the most efficient approach to estimating the gradient direction $\nabla g_j(X)$. Figure 5 illustrates 2^n factorial and n -simplex designs.

Biles [2,3] has described a first-order response surface procedure for approaching the constrained formulation of the multiple-response simulation problem. This procedure involves performing a first-order design around a current point X^k to estimate the gradient direction $\nabla g(X^k)$ according to relation (24). A line search is then performed along $\nabla g(X^k)$ to estimate an optimal step λ in (22). As long as the search remains interior to the region bounded by the constraints (11) or (12), the procedure is basically the same as that proposed by Box and Wilson [6]. If one or more constraints (11) or (12) are encountered, however, Biles [2,3] proposes that the gradient projection direction be followed (see Rosen [23]). The procedure for estimating the gradient projection direction is as follows.

Suppose that at an estimated boundary point X^k , q constraints are satisfied as equalities. These can be either the (11) or (12) constraints, or both. Let B_q be the $n \times q$ matrix of first partial derivatives of these active constraints. Thus, B_q consists of the q gradient vectors $\nabla g_j(X^k)$, $j=1, \dots, q$. That is,

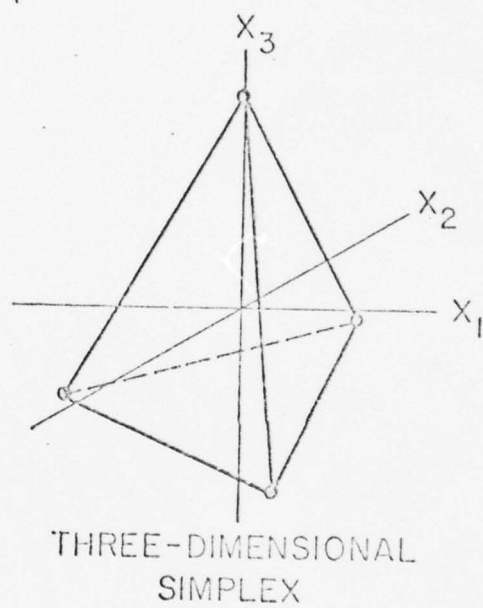
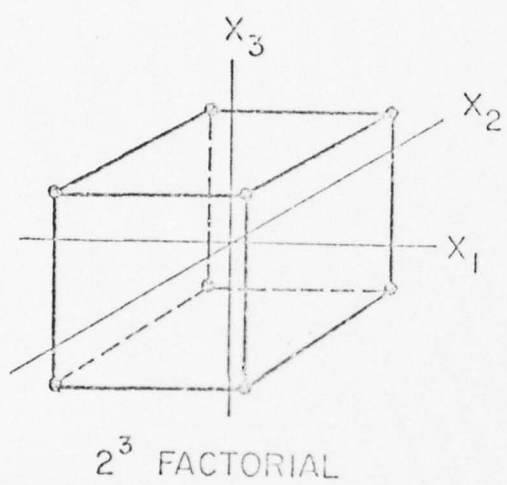


Figure 5. First-Order Response Surface Designs.

$$B_q = \begin{vmatrix} \partial g_1 / \partial x_1 & \dots & \partial g_q / \partial x_1 \\ \vdots & & \vdots \\ \partial g_1 / \partial x_n & & \partial g_q / \partial x_n \end{vmatrix} \quad (38)$$

Since $g_j(X)$, $j=1, \dots, q$ denotes the set of binding constraint functions (a constraint (11) or (12) is said to be "binding" if it is satisfied at the equality), for the moment let $f(X)$ represent the objective function. Then $\nabla f(X^k)$ and $\nabla g_i(X^k)$, $j=1, \dots, q$ represent the gradient vectors of the objective and constraint functions, respectively, evaluated at the boundary point X^k .

Performing a first-order response surface experiment about the boundary point X^k yields estimates of the gradient vectors $\nabla f(X^k)$ and $\nabla g_i(X^k)$, $j=1, \dots, q$ in the form of the vectors of regression coefficients. (If a constraint of type (11) is included in the set of binding constraints, the gradient vector has the form $(0, 0, \dots, 1, \dots, 0)'$, where all elements are zero except the i -th element which is one). Following the procedure outlined by Rosen [23], the gradient projection direction is given by

$$S^k = [\nabla f(X^k)] - B_q (B_q' B_q)^{-1} B_q' [\nabla f(X^k)] \quad (39)$$

A "golden section" line search is performed along direction S^k until either (a) a local "optimum" is found, or (b) other constraints are encountered. This new point is denoted X^{k+1} . This procedure is repeated until the gradient projection direction S^k is approximately zero. This point X^* is taken as a "constrained optimal" solution. Figure 5 illustrates the application of the gradient projection procedure to a constrained optimization problem.

Swain [27] has compared other first-order response surface techniques, including those of Klingman and Himmelblau [20] and Zoutendijk [28], to

Rosen's gradient projection method [23]. He found little difference among these techniques in terms of experimental requirements, and hence computer simulation time, but saw significant variability in computational requirements in order to use these algorithms. The Zoutendijk methods of feasible directions [28] require greater computational effort than the other procedures.

Biles [4] demonstrated both first-order and second-order approaches to a nonlinear goal programming formulation of the multiple-response simulation problem. These approaches are based on Ignizio's adaptation [17] of the method of Griffith and Steward [13] to goal programming and, like the constrained procedures mentioned previously, combine simulation, experimental design and mathematical programming.

The following generalized procedure is followed in employing a first-order response surface approach to the multiple-response simulation problem. The particular problem formulation and optimization procedure will govern the precise sequence of steps in implementing this procedure.

1. Identify the known experimental region $a_i \leq x_i \leq c_i, i=1, \dots, n$.
Selecting a starting point X^0 within this region. With X^0 as its center, array an orthogonal first-order response surface design within a selected design radius. Place $n_c = n/2 \geq 2$ points at the design center X^0 (coded as the $\bar{0}$ - vector).
2. Perform simulation trials at each of the N experimental design points and record the responses $y_j^\ell, j=1, \dots, m; \ell=1, \dots, N$. Using multiple linear regression, fit linear models of the form (23) and (25).
3. Apply the appropriate mathematical programming technique to locate the next center point in the search.
4. Repeat steps 1-3 until an "optimum" solution is located. It may be appropriate to add design points to complete a second-order response surface design to test this optimum solution. The procedure for

accomplishing this is described in the next section.

Figure 6 illustrates a sequential first-order response surface method applied to a constrained system. Figure 7 gives a flow chart for a simulation/optimization method based on the procedure outlined above. Appendix B gives a program listing for this procedure.

Second-Order Response Surface Methods

A second-order response surface approach to the multiple-variable, multiple-response simulation problem consists of one or more repetitions of a two-stage procedure: (a) the execution of a computer simulation trial at each point in a second-order response surface experimental design covering the known region given by (11) and the use of multiple linear regression to fit second-order regression models to the resulting data; and (b) the application of a suitable mathematical programming procedure to obtain a solution to the problem formulated in (10)-(12), in (15) together with (11), in (16) together with (11), or in (19)-(21). In contrast to the first-order methods, in which the optimization procedure was part and parcel with the experimental procedure, these procedures are distinct and sequential in the proposed second-order approaches.

The first step in the second-order approach is to identify the range of each input variable. A safe strategy is to cover the entire known region $a_i \leq x_i \leq c_i$, $i=1, \dots, n$ with the first (and possibly only) experimental design. If we let α_i denote the radius of the n -dimensional hypersphere within which the design points are contained, then

$$\alpha_i = (c_i - a_i)/2, \quad i=1, \dots, n \quad (40)$$

is effectively the maximum radius we could construct. It is convenient to adopt the coding convention expressed in (29) - (30), but choosing x_{iu} in such a way that α_i satisfies (40). Myers [22] describes this coding process.

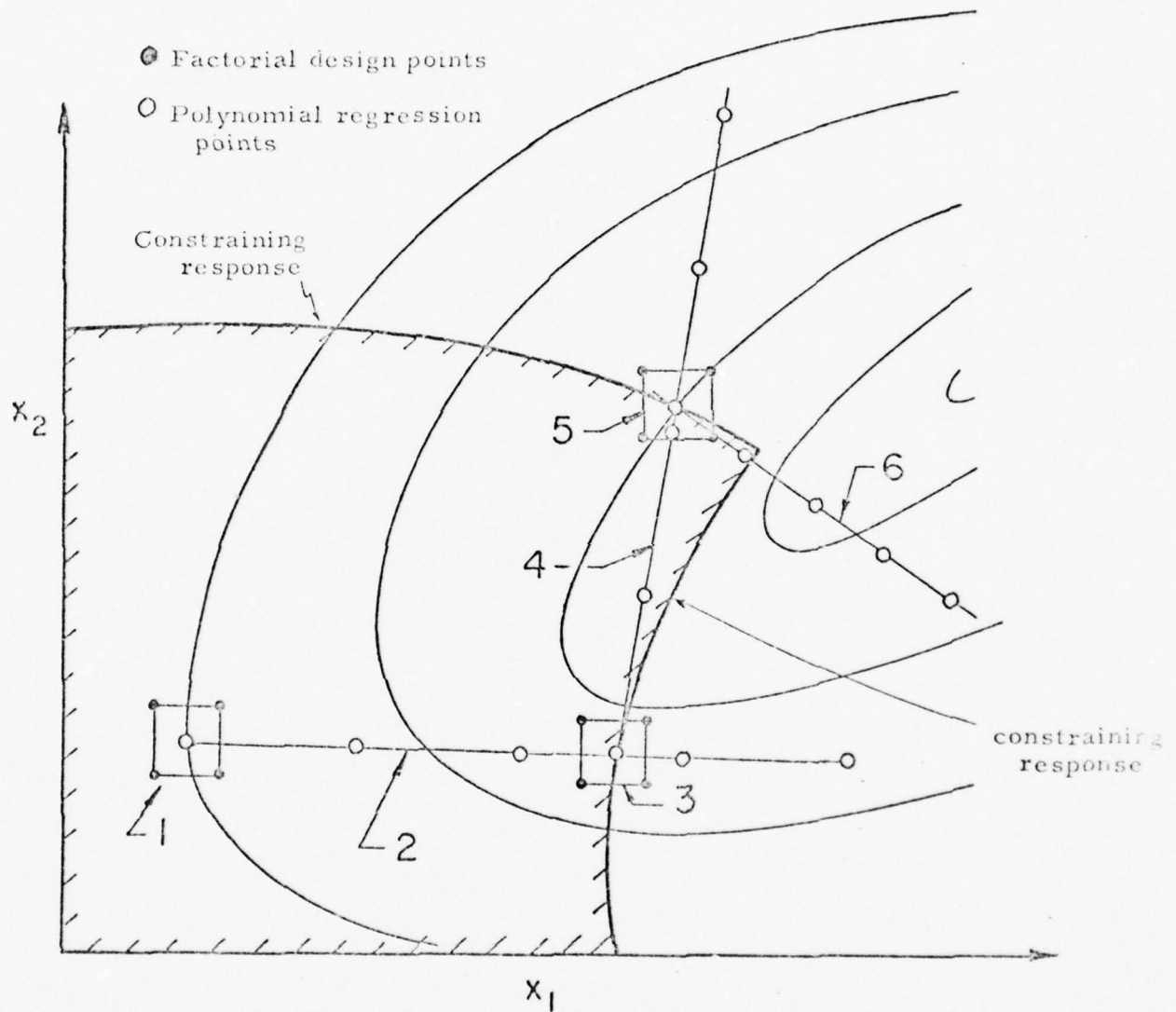


Figure 6. Gradient Projection Search.

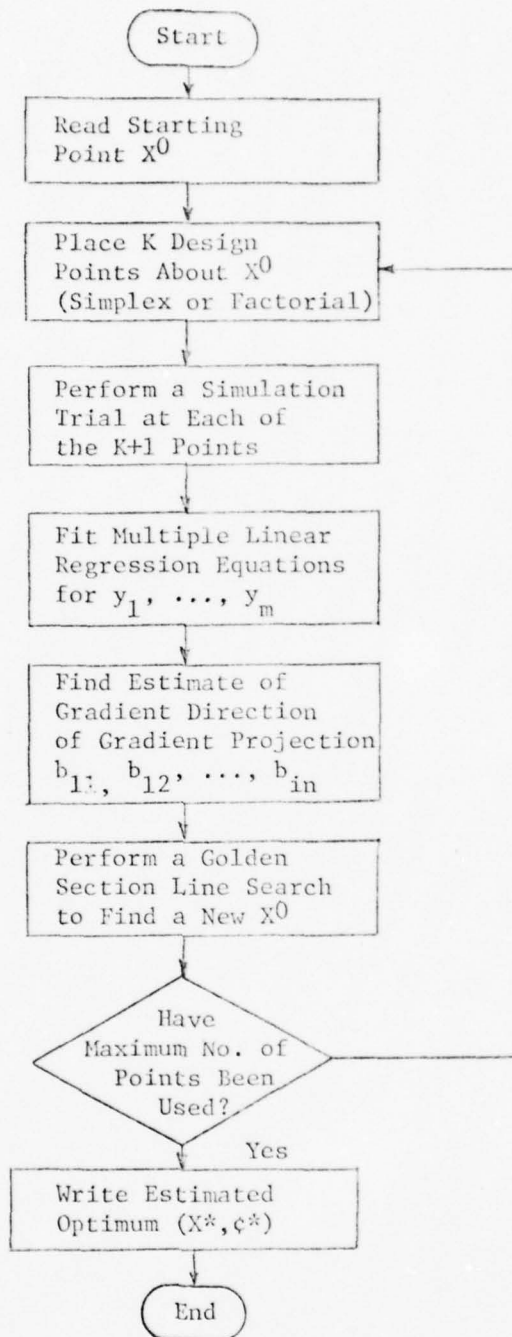


Figure 7. Flow Chart of Gradient Projection Search

The second-order fitted response surface has the form

$$\hat{y} = b_0 + \sum_{i=1}^n b_i x_i + \sum_{i=1}^n b_{ii} x_i^2 + \sum_{i=1}^n \sum_{j=1}^n b_{ij} x_i x_j \quad (41)$$

where \hat{y} is the estimate of the true response η at a given value

$X = (x_1, \dots, x_n)$ and the b_i and b_{ij} are regression coefficients in the fitted model. Since we must estimate m separate response relationships, equation (41) is modified to

$$\hat{y}_k = b_{k,0} + \sum_{i=1}^n b_{k,i} x_i + \sum_{i=1}^n b_{k,ii} x_i^2 + \sum_{i=1}^n \sum_{j=1}^n b_{k,ij} x_i x_j \quad (42)$$

$i \neq j$
 $k = 1, \dots, m$

Given the independence of the m responses, these m regression equations can be estimated independently from a set of $N \geq (n+1)(n+2)/2$ data points obtained by performing a simulation trial at each point in a second-order response surface design.

An experimental design employed for the purpose of estimating the regression coefficients in (42) must contain at least as many design points as there are coefficients b_i and b_{ij} in the fitted model, of which there are $(n+1)(n+2)/2$. Because of the non-linearity of (42), the experimental design must also have at least three levels of each controllable variable $x_i, i=1, \dots, n$. It is also desirable to have a design which is rotatable; that is, the predicted response \hat{y} at some point X is a function only of the distance from the design center to X and not a function of the direction.

The most widely used design for fitting a second-order model is the central composite design, shown in Figure 8 for $n=2$ and $n=3$. These designs consist of a 2^n factorial (or suitable fraction thereof), augmented by $2n$ axial points and k center points. A central composite design can be made

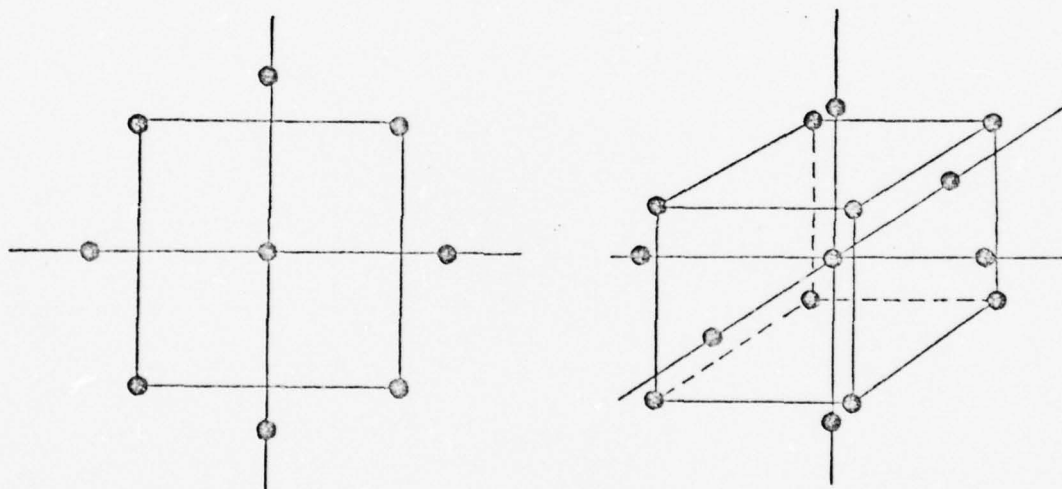


Figure 8. Central Composite Designs for Second-Order Response Surfaces.

rotatable by proper choice of α , the distance of the axial points from the design center. With the proper choice of the number of center points k , the central composite design can be made either orthogonal or uniform precision. Box and Hunter [7] give the characteristics of these designs for various sizes of n .

Another important class of second-order response surface designs is the equiradial designs. That is, $N \geq (n+1)(n+2)/2$ design points are placed at points on an n -dimensional hypersphere. Figure 9 illustrates two equiradial designs for two variables. It is important to note that each of these designs is also equiangular, in that the N points are placed in such a way as to form N equal angles from the design center. Biles and Swain [5] have demonstrated the construction of such designs based on orthogonal n -simplex designs, thus achieving an efficient rotatable design.

Having estimated the m second-order regression equations (42) and formulated the appropriate optimization problem, it remains to apply mathematical programming to obtain a solution. For the constrained formulation, any of the following procedures could be employed: (a) Box's complex search [8]; (b) Rosen's gradient projection method [23]; or (c) one of Zoutendijk's methods of feasible directions [28]. For the weighted objective function formulation, these same three procedures are applicable. For the goal programming formulation, Ignizio's [17] procedure based on the method of Griffith and Stewart [13] is computationally efficient.

Figure 10 illustrates a complex search applied computationally to second-order response surfaces. Figure 11 gives a flow chart for a simulation/optimization method which employs the computational version of Box's complex search applied to second-order regression equations estimated from data obtained from simulation trials performed at each point in a second-order

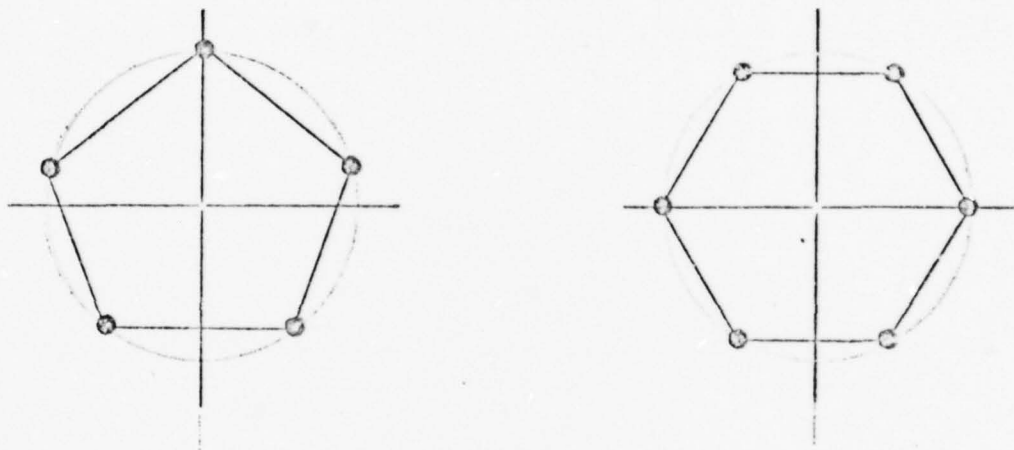


Figure 9. Equiradial Designs for Second-Order Response Surfaces.

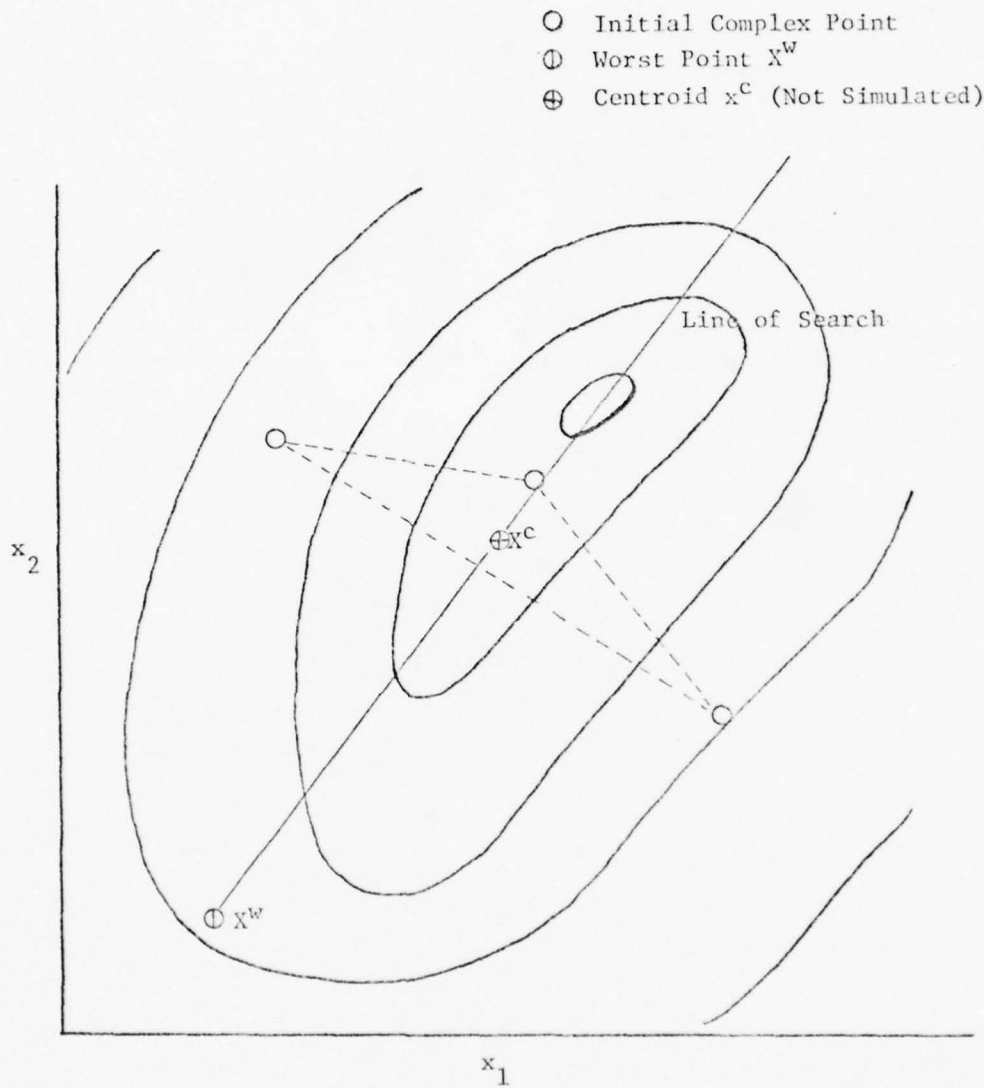


Figure 10. Complex Search with a Second-Order Surface.

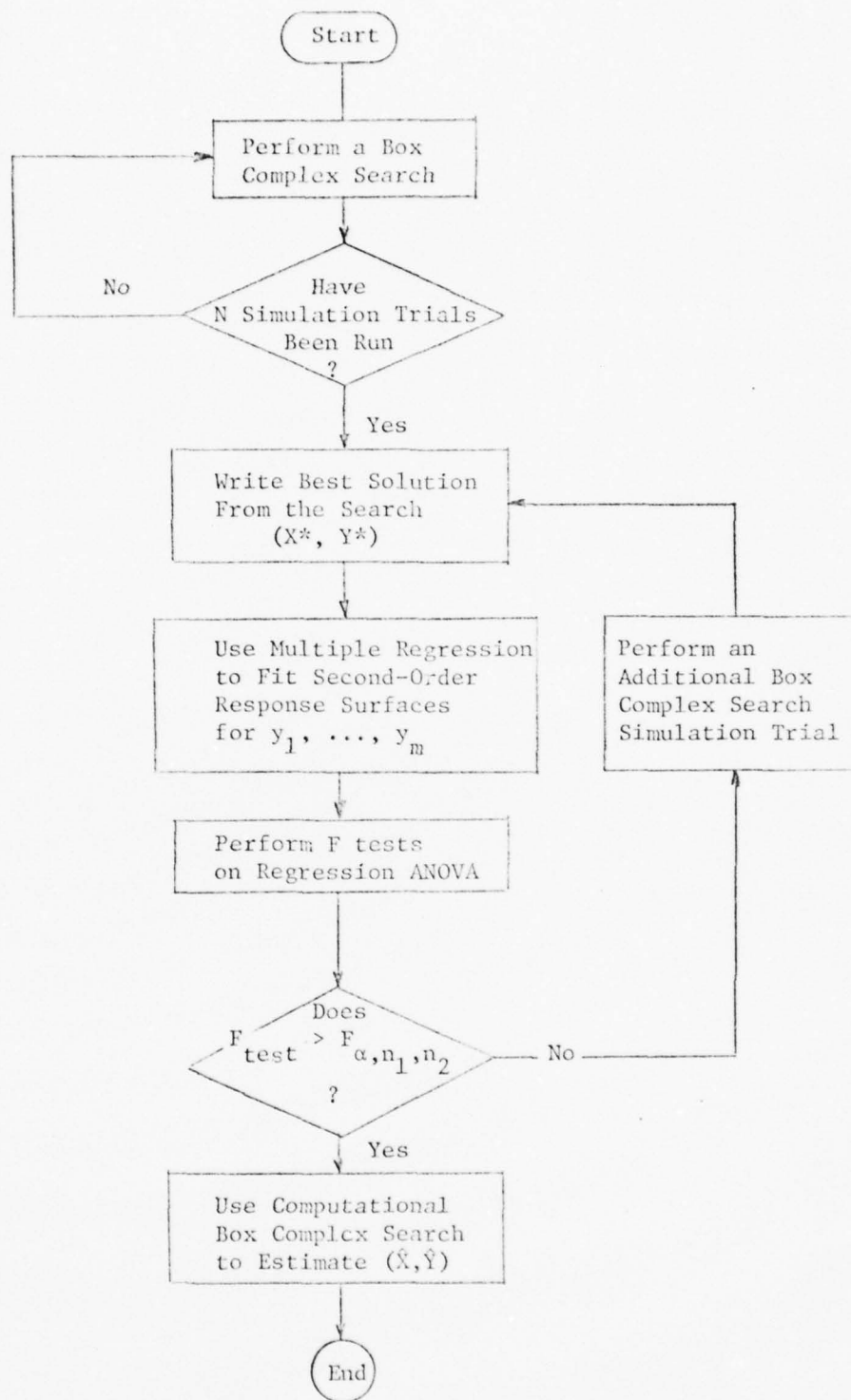


Figure 11. Flow-Chart of a Technique for Fitting Second-Order Response Surfaces to Complex Search Points

response surface design. Appendix C gives the program listing for this procedure.

Once an "optimal" solution has been obtained, it is necessary to compute a confidence interval, say at 90%, about the predicted "optimum." That is, we say that

$$P[y_{j\ell} \leq \eta_j^* \leq y_{ju}] = 0.9, j=1, \dots, m \quad (43)$$

where $y_{j\ell}$ and y_{ju} are lower and upper bounds, respectively, on the 90% confidence interval for response η_j^* at X^* . Draper and Smith [10] give the procedure for computing this confidence interval. If the range $(y_{ju} - y_{j\ell})$ is found to be excessively large, it may be necessary to perform confirmation simulations in the vicinity of the predicted "optimum," or even an entire repetition of the second-order design in a reduced experimental region about the predicted "optimum."

Second-Order Response Surface Analysis of Box's Complex Search Experiments

Another approach to the simulation/optimization problem is to combine second-order response surface analysis with Box's complex search. Such a procedure would be applicable to any of the problem formulations described earlier. The general procedure for this method is as follows:

1. Perform a set of Box's complex search experiments as described in the first procedure.
2. Using the simulation responses at each search point, fit second-order regression equations.
3. Apply Box's complex search computationally to these equations.

The advantages of this two-phase procedure over either of its "component" techniques are that (1) it provides an objective stopping mechanism for the experimental Box complex search, and (2) it extracts latent information

about the unknown response functions $\eta_j(X)$, $j=1, \dots, m$ that the search alone would not discover. The following is a description of this simulation/optimization procedure.

The Box's "complex" search phase of simulation experimentation proceeds as follows:

1. Randomly generate a set of $N \geq n + 2$ search points satisfying the known bounds $a_i \leq x_i \leq c_i$, $i=1, \dots, n$.
2. Perform a simulation trial at each of these N search points X^k , $k=1, \dots, N$ and record the mN simulated responses y_j^k , $j=1, \dots, m$, $k=1, \dots, N$.
3. Where a given search point X^k is found to violate a constraint of the form

$$y_j^k \approx \eta_j(X^k) \begin{matrix} > \\ = \\ < \end{matrix} d_j, \quad j=1, \dots, m \quad (6)$$

generate a replacement point $X^{k'}$, perform a simulation trial at $X^{k'}$, and assess the feasibility of these new responses $y_j^{k'}$, $j=1, \dots, m$. Repeat this step until exact N search points satisfying both the known bounds and any constrained responses (6) are obtained.

4. Identify as X^w that search point yielding the least desirable set of responses y_j^w , $j=1, \dots, m$. Compute the centroid X^c of the $N-1$ remaining search points. Project X^w through the centroid X^c to a new point. If this point satisfies the known bounds, perform a simulation trial to obtain the response $y_j^{w'}$, $j=1, \dots, m$ at $X^{w'}$. If any responses fail to satisfy the constrained responses (6), shorten the step from X^w through X^c , and repeat this step until a feasible new point $X^{w'}$ is obtained. This point replaces X^w in the "complex".

5. Repeat step 4 until either of the following conditions occurs:

- (a) an estimated solution is obtained at (\hat{X}, \hat{Y}) ; or
- (b) a predetermined number of search points K (both those satisfying (6) and those not), is obtained, where

$$K \geq (n+1)(n+2)/2.$$

When the complex search phase of simulation experimentation has been terminated, a best available, feasible solution (\hat{X}, \hat{Y}) is retained for future reference. An examination is made of the arrangement of search points X^k , $k=1, \dots, K$ in the known experimental region $a_i \leq x_i \leq c_i$, $i=1, \dots, n$ to detect any regions that have a sparse density of search points. Any such region will have a search point placed at or near its center and a simulation trial performed there. This process guarantees coverage of the entire experimental region. It is also worthwhile to compute the centroid of the T total search points and to perform R replicates of a simulation trial (with different sets of initial random number seeds) at this centroid point. This total set of $W=T+R$ simulation trials comprises a random experimental design in the factor space $a_i \leq x_i \leq c_i$, $i=1, \dots, n$ with something approaching the minimum number of points for estimating second-order response surfaces. This minimum number of points is $(n+1)(n+2)/2$, corresponding to the number of regression coefficients in the model (42). Multiple linear regression is then employed to fit each of the m second-order surfaces to the W sets of responses y_j^k , $j=1, \dots, m$, $k=1, \dots, W$. A Box's complex search procedure is applied computationally to this system of m second-order response surfaces according to the particular problem formulation. The solution obtained by this procedure (X^*, Y^*) is compared to that obtained by complex search (\hat{X}, \hat{Y}) . If (X^*, Y^*) is preferred to (\hat{X}, \hat{Y}) and X^* and \hat{X} differ significantly, one or more confirmation simulation trials must be performed at (X^*, Y^*) and possibly

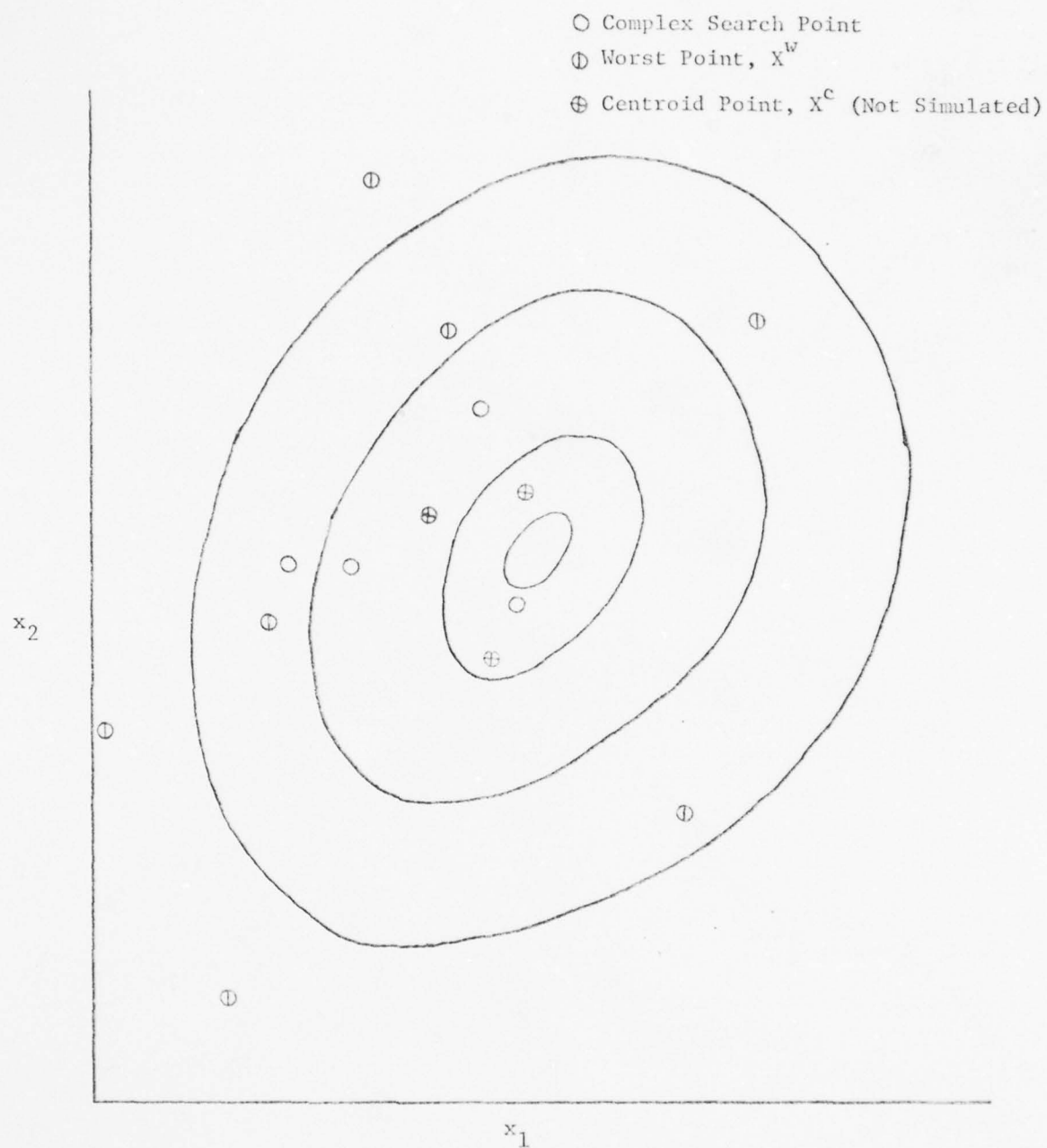


Figure 12. Second-Order Surface Fitted to a Sequence of Complex Search Points.

at (\hat{X}, \hat{Y}) as well to enable a choice of the "optimum" solution.

Figure 12 illustrates a second-order response surface fitted to a set of points obtained via an experimental complex search, and then a computational complex search applied to this surface. Figure 11 gives a flow chart for this simulation/optimization procedure. Appendix D is a program listing of this procedure.

SIMULATION TEST MODELS

Three simulation test models were used to evaluate each of the four optimization methods. These simulation models are as follows:

1. A stochastic inventory system as described in Ignall [16] and Hunter and Naylor [15].
2. A tank duel model as described in Montgomery and Bettencourt [21].
3. A naval minefield evaluation model as described by Bailey and Weedon [1].

Exploratory work was performed but not completed on a fourth simulation model, the SPEARS anti-aircraft Naval defense model by Kaplan et al [18]. These models are discussed in detail in the following sections, but test results are discussed later in this report.

Stochastic Inventory System

Ignall [16] and Naylor and Hunter [15] describe the application of experimental design to the optimization of computer simulation responses. They employed as a simulation test model a discrete-event simulation of a stochastic inventory system in which mean daily demand and order lead time are random variables with known probability density (mass) functions. The lone simulation response was

$$y = \text{mean daily cost, \$}$$

which was the sum of carrying, setup and shortage costs. The two controllable variables were

$$x_1 = \text{reorder point (ROP) ,}$$

$$x_2 = \text{economic order quantity (EOQ),}$$

The optimization problem was

$$\text{minimize } y = n(x_1, x_2)$$

subject to

$$-5 \leq x_1 \leq 90$$

$$50 \leq x_2 \leq 250$$

Ignall [16] found as a solution $y = \$76$ at $x_1 = 45$ and $x_2 = 175$ units, for a given set of values for the several constants in the model.

The manner in which this simulation test model was utilized for the current research did not involve an actual simulation program. Rather, the 20 x 9 response table was used as the simulator. This response table is shown in Figure 13. For a given (x_1, x_2) , an interpolation was performed in this response table to find y . This enabled a comparison of the four optimization procedures without introducing a confounding influence from the simulation model. It should also be noted that, since the "simulation" produced a single response y , this simulation test model did not fully test the optimization procedure but principally evaluated the interface between the simulator and optimization modules.

Tank Duel Model

In describing the application of multiple response surface methods in computer simulation, Montgomery and Bettencourt [21] employed a stochastic simulation model of a tank duel. The model simulates brief fire engagements between two armored vehicles. A stationary defending vehicle (Blue Tank)

		y, AVERAGE DAILY COST, \$								
		x_2 , EOQ								
x_1 , ROP		50	75	100	125	150	175	200	225	250
-5		312	221	168	143	123	118	107	104	100
0		288	202	155	130	112	109	101	97	96
5		267	185	143	120	105	101	96	82	92
10		249	170	132	110	99	97	91	88	88
15		232	159	125	105	95	91	89	88	87
20		215	147	117	97	90	87	86	83	85
25		198	136	109	92	85	83	84	81	84
30		180	124	100	88	80	80	81	79	83
35		167	117	93	85	78	78	79	80	83
40		160	111	91	84	77	77	79	79	82
45		154	107	90	81	78	76	79	81	83
50		145	102	87	80	77	77	79	81	83
55		142	101	86	79	78	79	80	82	85
60		138	100	86	79	79	79	80	85	87
65		135	99	85	79	80	80	81	85	88
70		134	100	86	81	81	81	83	87	90
75		134	101	88	83	83	84	85	89	93
80		135	101	89	85	84	86	87	90	95
85		134	103	91	87	86	88	88	92	97
90		135	105	93	88	88	89	91	95	99

Figure 13. Response Table for Stochastic Inventory Simulation (Ignall [16]).

fires first at a fully-exposed attacking vehicle (Red Tank). The engagement ends when a kill occurs or a predetermined time limit of 120 seconds expires.

The input variables to the tank duel model are presented in detail in [21]. But for purposes of evaluating and comparing the four simulation/optimization procedures described in this report, the following independent variables x_i and response variables y_j were chosen:

x_1 = mean time to fire first round for the Blue crew (sec.),

x_2 = mean time between rounds (sec.),

y_1 = probability of Blue victory,

y_2 = expected number of rounds fired by the Blue tank.

The optimization problem was framed as one of constrained optimization, as follows:

$$\text{maximize } y_1 = g_1(x_1, x_2)$$

$$\text{subject to } 5 \leq x_1 \leq 25 \quad 5 \leq x_2 \leq 25 \quad 0 \leq y_2 \leq 2$$

Mine Hunting [1]

This model, written in FORTRAN IV, is a Monte Carlo computer simulation which is used to evaluate mine hunting tactics. This program may be used either to evaluate minefields or, in conjunction with a sustained attrition minefield evaluation model, to compare the effectiveness of minesweeping to that of mine hunting. The basic operation is to move a mine hunting ship through a minefield. Each time the ship passes a mine or minelike object, its distance from that object is calculated, and a random number in the (0,1) range is drawn and compared with the probability of the ship's sonar detecting the object. If the object is detected, a similar procedure determines if the object is correctly classified and, in the case of mines, neutralized.

The ship's course is altered to avoid mines it has detected. Actuation and detonation of mines and damage of ships is simulated for both mine hunters and traffic ships.

Inputs include the number of mines, shape of minefields, density of bottom clutter, number of hunters, length of operation, effectiveness of sonar, condition of bottom, number of traffic ships, and method of hunting. Geographically, the model is limited to an area 20 miles square (a function of the number of mines) while the force limits are 100 hunters, 5 traffic ship types, 60 types of mines or minelike objects (e.g., bottom clutter), and 750 individual mines or minelike objects.

Outputs include mines detected, mines neutralized, ship damage, and threat of the minefield as a function of time.

Two separate problems were used to evaluate the four optimization procedures with the mine hunting model. In each of these problems, the optimization was sought from the standpoint of the force which deploys the minefield. That is, the objective was to maximize minefield effectiveness in terms of damage to ships which attempt to traverse the field.

Problem 1 used two mine types, arming delays and a total of 58 mines. Problem 2 used only one mine type, no arming delays and 61 mines. In Problem 1, all traffic ships (i.e., the target ships as opposed to the mine sweepers) attempted to transit the mine field during the period 119-120 hours, while in Problem 2 traffic was uniform over the entire 120 hour period.

Additional information about the problem that was common to both plans is as follows:

- The minefield was to cover an area one nautical mile wide and ten nautical miles long. (The input provided was for a 1800 ft. x .0 nm. field; this area required fewer mines, hence reduced running time. It was assumed that mine requirements were proportional to the area

- of the minefield so that the plan for a 1nm. x 10 nm. area used $(6076.1/1800) \times$ the number of mines in the 1800' x 10 nm. area).
- Water depth was constant (hence for a given mine type/target type/target speed combination, the characteristics of a mine-target interaction were the same everywhere).
 - The target ships travelled at three knots with a navigational error of 100 yards (i.e., the perpendicular distance from the target's actual location to the intended location was normally distributed with a mean of zero and a standard deviation of 100 yards).
 - Two minesweepers were available and had a combined capability of making ten transits per day at a safe speed of five knots. They were assumed to have actuation characteristics identical to the target but were safe from damage.
 - The mines were assumed to be uniformly randomly distributed throughout the minefield.

The objective of each plan was to provide a minefield with a 90 percent chance of damaging (mission about level) at least two of the ten target ships.

The simulation/optimization problem for each of these two minefield evaluation problems was formulated as follows:

- x_1 = mean of the arming time distribution, where arming time was Poisson distributed; $6 \leq x_1 \leq 12$ hrs.
- x_2 = mean of the mine count distribution, where mine count was Poisson distributed; $3 \leq x_3 \leq 5$ counts.
- y_1 = average ship damage, ships (summed over all ship types)
- $y_j, j = 2, \dots, 6$ = average "threat" in period $j-1$, where period 1 is 0 - 24 hrs., period 2 is 24-48 hrs., etc.

A constrained optimization problem was

$$\max y_1$$

subject to $6 \leq x_1 \leq 12$; $3 \leq x_2 \leq 5$; $0 \leq y_j \leq 0.25$, $j = 2, \dots, 6$.

The simulation/optimization interface for the minefield evaluation model involved the following:

- The main program was modified to become Subroutine SIMUL in the optimization program.
- In Subroutine SIMUL, each mine was assigned random arming time and a random ship count from Poisson distributions having parameters x_1 and x_2 , respectively.
- In Subroutine AVGTHR, in which the damage to traffic ships is tallied and minefield "threat" is computed, special program segments were written to give values to the response variables y_j , $j = 1, \dots, 6$.

Air Attack on a Surface Fleet [18]

This model, written in FORTRAN IV, evaluates antiair warfare tactics related to the following factors:

- i. Force disposition - e.g., dispersed or integral formation concepts;
- ii. Fire control doctrine - e.g., shoot, look, shoot;
- iii. Command and control - e.g., coordinated or autonomous assignment doctrine;
- iv. Cover and deception - e.g., "look alike" ships and decoy forces;
- v. Deployment of remote sensors - e.g., antielectronic warfare aircraft and radar picket ships;

The effects of enemy force tactics evaluated include multidirectional, multialtitudinal attacks, use of electronic countermeasures and penetration aids, "roll back" attack tactics, and weapon loading and launch criteria.

The model may be described as a one-sided, event-store, Monte Carlo computer simulation. Enemy attack tactics are fully defined at the outset of each game situation by inputs to the raid generator, and the attack is conducted as defined regardless of the evolution of the battle. Thus, enemy attack tactics fall into the category of the "uncontrollable factors" z_k , $k = 1, \dots, p$ in equation (1). The defense force, typically a carrier task force, includes a surveillance radar network (shipborne and airborne air search radars, both conventional periodic scan types, and multimode array radars), a communications network for track passing and firepower coordination, and defensive weapon systems (primarily SAM batteries and their associated tracking radars and fire-control computers). Task force runs may be made to establish sensitivity to changes in ship configurations (alternate radars or weapon systems) or in ship dispositions and assignment doctrine. The model was designed to be, to the greatest extent practicable, machine and installation independent.

Input data to this model define the size, composition, disposition, and fire-control tactics for both the offense and the defense units. Enemy forces are most often composed of aircraft and submarines that launch antiship cruise missiles from stand-off ranges, although other forms of attack are possible (e.g., gravity bombs). The model has the capability of handling up to 31 defense units (ships and AEW aircraft) and up to 255 offense units (launch vehicles, stand-off jammers, cruise missiles, etc.). Numerous other limits define the composition of each major unit (e.g., radars per ship, missile launchers per ship, magazine capacity per launcher, cruise missiles per launch platform, flight path legs per target). Certain input parameters select tactical options (e.g., single or dual missile salvos) and simulation algorithm options (e.g., simple or complex radar detection models).

A summary of the processed input values is printed at the outset of a game situation. This may optionally be followed by a detailed listing of

event occurrences (of all types or of a selected subset and over the entire battle or during a specified time interval). A summary listing at the completion of play tabulates the performance and status of individual units at the end of the game; this listing includes factors such as average range of detection by surveillance radars, intercept ranges and kill probabilities achieved by defensive missile systems, number of enemy weapon penetrations (hits) suffered by each defense unit, and damage status of major ship components at the end of game play.

Although no simulation/optimization trials were conducted with this model, trial simulations were executed to characterize its performance. As with the minefield evaluation model, the "air attack on a surface fleet" model would require that the optimization interface be "custom-fitted" by developing suitable FORTRAN code.

DISCUSSION OF RESULTS

This section discusses the results obtained from simulation/optimization runs with the four optimization procedures in combination with each of the three simulation models. The discussion is presented in three segments, one for each simulation model.

Stochastic Inventory Model [15,16]

Table 1 presents the results of several optimization attempts with the stochastic inventory model. The complex search procedure and the second-order response surface approach of employing multiple regression to fit a quadratic model to a set of complex search points gave very similar results, yielding near-optimal solutions with about $7n$ simulation trials. The second-order response surface method employing a central composite design produced a predicted optimum at a point for which the actual solution was 4 percent removed from the true optimum, using only 9 simulation trials. The sequential first-order response surface approach, employing either a 2^2 factorial design or a 3-vertex simplex design (each augmented by a centroid point), with a golden section search along the gradient direction, required about $20n$ simulation trials to yield near-optimal solutions. Among the four optimization methods evaluated with this model, only the first-order response surface method fared poorly.

It is notable that, when a second-order response surface is fitted to the 14 search points obtained from a Box complex search, the predicted optimum solution is actually slightly worse than the best solution obtained via the search itself.

Table 1. Summary of Simulation/Optimization Results
for the Stochastic Inventory Model [15,16]

Optimization Method	Starting Seed	Estimated Solution			Number of Trials
		x_1	x_2	y	
Complex Search	12471	50	177	\$76.20	13
	21437	43	245	81.00	14
First-Order Factorial Design	17332	70	125	\$79.00	40
First-Order Simplex Design	17332	42	159	\$76.78	41
Second-Order Central Composite Design	35188	53	183	\$70.32 (Predicted)	9
				\$79.00 (Actual)	
Second-Order Complex Search	14271	41	163	\$76.78 (Search)	14
				77.52 (Predicted)	

Notes: (1) Known Solution $x_1 = 45$ $x_2 = 175$ $y = \$76.00$

(2) CPU times less than 5 sec. per optimization run on
an IBM 370/168 computer.

(3) Complex search terminated because $2n=4$ search points
were conducted without an improved solution.

Tank Duel Model [21]

Table 2 presents the results of four simulation/optimization runs with the tank duel model. An optimum solution lies near ($x_1 = 8.2$, $x_2 = 12.5$), Producing a probability of Blue Victory of 0.61 with an expected firing of 2 Blue rounds. Thus, the "expected rounds fired" response appears to bound the solution. As with the stochastic inventory model, only the sequential first-order response surface approach produces unacceptably costly experimentation.

In contrast to the result with the stochastic inventory model, the predicted solution obtained after fitting second-order response surfaces to the two sets of responses obtained in the Box complex search represents a significant improvement over the best point observed in the search.

Mine Hunting Model [1]

As seen in Table 3, only three optimization methods were evaluated with the mine hunting simulation model. The complex search method, as with the previous models, was terminated after 2n simulation trials had been performed without obtaining an improved solution. For example, with problem 1, the estimated optimum solution was observed with search point 12, and since four more search points failed to improve on that result the search was terminated after point 16.

Focusing on the results of the second-order response surface method, employing a central composite design with 16 simulation trials, a solution $x_1 = 10.6$ hours mean arming delay per mine and $x_2 = 3.1$ ships mean ship count per mine results in an expected kill of 0.51 ships. The threat profile is essentially zero during the first 72 hours of minefield operation, and increases to levels between 0.15 and 0.2 during the last 48 hours of operation when enemy ship traffic actually attempts to traverse the field.

Table 2. Summary of Simulation/Optimization Results
for the Tank Duel Model [21]

Optimization Method	Estimated Solution				Number of Trials
	x_1	x_2	y_1	y_2	
Complex Search	12.9	10.1	0.59	1.97	20
First-Order Simplex Design	8.2	12.4	0.60	1.97	55
Second-Order Central Composite	8.0	12.5	0.61	2.00	9
Second-Order	13.5	11.3	0.57	1.86 (Search)	15
Complex Search	8.2	12.2	0.61	1.99 (Predicted)	

- Notes:
- (1) Known optimum at $x_1 = 8.2$ sec., $x_2 = 12.5$ sec., $y_1 = 0.61$, $y_2 = 2$ rds.
 - (2) CPU times less than 5 sec. per optimization run on an IBM 370/168 computer.
 - (3) Complex search was terminated because $2n = 4$ search points were evaluated without obtaining an improved solution.

Table 3. Summary of Simulation/Optimization Results
for the Minefield Evaluation Model [1]

Optimization Method	Problem Number	x_1	x_2	y_1	y_2	y_3	y_4	y_5	y_6	Number of Trials
Complex Search	1	11.4	3.9	0.43	0.000	0.000	0.012	0.014	0.030	16
	2	6.2	3.5	0.32	0.000	0.027	0.027	0.044	0.058	9
Second-Order Central Composite	1	10.6	3.1	0.51	0.000	0.000	0.000	0.183	0.163	16
	2	6.0	3.0	0.37	0.000	0.005	0.044	0.091	0.128	16
Second-Order Complex Search	1 (Search)	8.6	3.6	0.55	0.000	0.000	0.000	0.183	0.140	25
	1 (Predicted)	8.8	4.0	0.26	0.000	0.000	0.000	0.153	0.129	

Notes: (1) True optimum solution is unknown.

(2) In each simulation trial, 5 replicates of the simulation were averaged.

(3) Optimization runs required about 10 minutes of CPU time each on an IBM 370/168 computer.

(4) The sequential first-order response surface method was not evaluated due to excessive computer time.

(5) The second-order complex search procedure did not coverage in a 10-minute run.

This result points out a difficulty in interfacing an optimization procedure with a simulation model when a gaming strategy is involved. In a sense, the optimization procedure "learns" the enemy tactic as the series of simulation runs progresses. In this instance, the optimization procedure "sees" that traffic ships traverse the field late in the game. To counter this "learning" behavior, it is necessary to level the threat over the entire period of minefield operation. Since such a narrow feasible region would cause complex search to yield a high proportion of infeasible responses, the preferred optimization method would be a second-order response surface approach, either by a designed experiment or by fitting a quadratic surface to Box's complex search results.

SUMMARY AND CONCLUSIONS

This research has investigated four alternative approaches to optimization of simulated systems having multiple independent variables x_i , $i=1, \dots, n$ and multiple simulation responses y_j , $j=1, \dots, m$. Of these four methods, Box's complex search [8], a second-order response surface approach employing a central composite experimental design [6,7] followed by a computational complex search, and fitting a second-order response surface to a succession of complex search points, all yield "good" solutions in an "economical" number of computer simulation trials. The sequential first-order response surface method converged to near-optimal solutions, but required an excessive number of simulation trials. This excessive number of trials derived from the golden section line search along the estimated gradient direction. Had a polynomial regression line search employing, say, five trials been conducted as in Biles [2,3], two complete cycles of gradient-determining blocks and line searches could have been performed in about $8n$ to $10n$ trials.

The most crucial aspects of the experimental Box complex search approach to simulation/optimization are twofold:

1. A termination criterion based on either (a) a maximum number of simulation trials or (b) a certain number of simulation trials without obtaining an improved solution. The typical criterion used to terminate a computational Box complex search is convergence; such as $y_{k+1} - y_k \leq \epsilon$. Such a criterion can lead to excessive simulation trials in the experimental mode, however, unless ϵ is coarse.
2. In an experimental region in which the implicit constraints on simulation responses y_2, \dots, y_m describe a small feasible region, a number of infeasible simulation trials will be conducted. In fact, the probability that an early simulation trial is feasible with respect to implicit constraints is $P_f = R(Y_f)/R(E)$, where

P_f = probability of a feasible trial

$R(Y_f)$ = region subtended by the implicit constraints

$R(E)$ = region subtended by the explicit constraints

Once $n+2$ feasible simulation trials have been found, and complex search begins to seek improved solutions, there is a higher probability of obtaining a feasible simulation trial. As the "complex" becomes tighter, this probability approaches one. But the possibility of a large number of infeasible trials early in the search is perplexing.

But the capability of fitting quadratic response surfaces to a set of at least $(n+1)(n+2)/2$ complex search points (this being the number of regression coefficients in the quadratic model) in an "incomplete" complex search overcomes this difficulty.

The second-order response surface approach has the advantage of requiring relatively few simulation trials, but the accuracy of the predicted optimum solution (\hat{X}, \hat{Y}) depends on how well the true surfaces are represented with quadratic estimators. In those cases where there is some knowledge of the nature of the true surfaces, the modeler is better able to hypothesize the fitted model (which may not be quadratic). Although the central composite design has superior statistical properties to the purely random design, such as obtained from a set of complex search points, the complex search approach will usually place more than one simulation trial in the neighborhood of the predicted optimum. This affords a smaller confidence region about the predicted optimum.

The computer programs listed in Appendices A through D offer a means of interfacing these optimization procedures with simulation models of interest. Minor alterations in the optimization program would be necessary for a particular simulation model, but major modifications in the simulation model itself could be entailed. In general, one could expect to have to "custom fit" the optimization program to a given simulation model. This is especially true of the typical naval simulation model, because modelers tend to "start from scratch" in developing models for their own needs.

This research has provoked two recommendations relative to naval simulation modeling activities in the several naval laboratories:

1. There is a definite need to have naval simulationists apply up-to-date statistical methodology, particularly variance reduction, in their modeling efforts.
2. There is a need to have naval simulationists become familiar with general simulation languages such as GASP-IV and SIMSCRIPT 2.5. It is also apparent that certain specialized modeling techniques, such as Q-CERT, is vastly under utilized in naval operations modeling.

Developing FORTRAN simulation models from "scratch" is sometimes necessary, but very often considerable time, effort and expense is spent in duplicating the kinds of functional capability contained in available and tested simulation software.

The most pressing need in terms of continuing research in the area of optimization of naval simulation models is to incorporate the techniques of game theory into the optimization approach. So many naval models embody a "friendly force/enemy force" focus, with decision-making capability on both sides, that optimization must entail game theory considerations.

REFERENCES

1. Bailey, J. L., and M. A. Weedon, "The Minefield Analysis with Hunting Evaluation Model (MAYHEM)," NWL Technical Report TR-2742, U. S. Naval Weapons Laboratory, Dahlgren, Virginia, May 1972.
2. Biles, W. E., "A Gradient-Regression Search Procedure for Simulation Experimentation," Proceedings of the 1974 Winter Simulation Conference, Washington, D.C., 1974, p. 491-497.
3. Biles, W. E., "Response Surface Methods for Optimization of Constrained Multivariable Processes," Industrial and Engineering Chemistry: Process Design and Development, Vol. 14, No. 2, 1975, p. 152-158.
4. Biles, W. E., "Optimization of Multiple-Objective Computer Simulations: A Nonlinear Goal Programming Approach," Proceedings of the 1977 AIDS Conference, Chicago, 1977.
5. Biles, W. E., and J. J. Swain, "Strategies for Optimization of Multiple Response Simulation Models," Proceedings of the 1977 Winter Simulation Conference, Gaithersburg, Maryland, December 1977.
6. Box, G. E. P., and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," Journal of the Royal Statistical Society, B, Vol. 13, 1951, p. 1-45.
7. Box, G. E. P., and J. S. Hunter, "Multifactor Experimental Designs for Exploring Response Surfaces," Annals of Mathematical Statistics, Vol. 28, 1957, p. 192-242.
8. Box, M. J., "A New Method of Constrained Optimization and a Comparison with Other Methods," Computer Journal, Vol. 8, 1965.
9. Brooks, S. H., and M. R. Mickey, "Optimization Estimation of Gradient Direction in Steepest Ascent Experiments," Biometrics, Vol. 17, No. 1, 1961.
10. Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley and Sons, New York, 1966.
11. Dyer, J. S., "Interactive Goal Programming," Management Science, Vol. 19, No. 1, 1973.
12. Fishman, G. S., Concepts and Methods in Discrete Event Digital Simulation, John Wiley and Sons, New York, 1973.
13. Griffith, R. E. and R. A. Stewart, "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems," Management Science, Vol. 7, 1961.
14. Hooke, R. and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems," Journal of the Association of Computing Machinery, Vol. 8, 1961.

15. Hunter, J. S., and T. H. Naylor, "Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 16, No. , 1970, pp. 422-434.
16. Ignall, E. J., "On Experimental Designs for Computer Simulation Experiments," Management Science, Vol. 18, No. 7, March 1972, pp. 384-388.
17. Ignizio, J. P., Goal Programming and Extensions, Lexington Books, D.C. Heath and Company, Lexington, Mass, 1976.
18. Kaplan, D. J., L. C. Davis, M. E. B. Owens II, O. F. Forsyth and J.J. Penick, "SPEARS, an AAW Performance Simulation," NRL Report 7958, Naval Research Laboratory, Washington, D. C., June 1976.
19. Kleijnen, J. P. C., Statistical Techniques in Simulation, Parts I and II, Marcel Dekker, New York, 1975.
20. Klingman, W. R., and D. M. Himmelblau, "Nonlinear Programming with the Aid of the Multiple Gradient Summation Technique," Journal of the Association of Computing Machinery, Vol. 11, 1964.
21. Montgomery, D. C., and V. M. Bettencourt, "Multiple Response Surface Methods in Computer Simulation," Simulation, October, 1977, pp. 113-121.
22. Myers, R. H., Response Surface Methodology, Allyn and Bacon, Boston, Mass., 1971.
23. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming: Part II - Nonlinear Constraints," Journal of the Society of Nonlinear and Applied Mathematics, Vol. 8, 1961.
24. Smith, D. E., "Requirements of an Optimizer for Computer Simulations," Naval Research Logistics Quarterly, Vol. 20, No. 1, 1973.
25. Smith, D. E., "An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation," Operations Research, Vol. 21, 1973, pp. 475-497.
26. Spendley, W., and G. R. Hext, and R. F., Himsworth, "Sequential Application of Simplex Designs in Optimization and Evolutionary Operations," Technometrics, Vol. 4, 1962.
27. Swain, J. J., "Efficient Strategies for Constrained Experimental Optimization," unpublished Master's thesis, University of Notre Dame, 1977.
28. Zoutendijk, G., Methods of Feasible Directions, Elsevier, New York, 1960.

APPENDIX A

PROGRAM LISTING

FOR AN

EXPERIMENTAL BOX COMPLEX SEARCH


```

12 10000 (12,120) 116X
13 10000 (12,120) 116X
14 10000 (12,120) 116X
15 10000 (12,120) 116X
16 10000 (12,120) 116X
17 10000 (12,120) 116X
18 10000 (12,120) 116X
19 10000 (12,120) 116X
20 10000 (12,120) 116X
21 10000 (12,120) 116X
22 10000 (12,120) 116X
23 10000 (12,120) 116X
24 10000 (12,120) 116X
25 10000 (12,120) 116X
26 10000 (12,120) 116X
27 10000 (12,120) 116X
28 10000 (12,120) 116X
29 10000 (12,120) 116X
30 10000 (12,120) 116X
31 10000 (12,120) 116X
32 10000 (12,120) 116X
33 10000 (12,120) 116X
34 10000 (12,120) 116X
35 10000 (12,120) 116X
36 10000 (12,120) 116X
37 10000 (12,120) 116X
38 10000 (12,120) 116X
39 10000 (12,120) 116X
40 10000 (12,120) 116X
41 10000 (12,120) 116X
42 10000 (12,120) 116X
43 10000 (12,120) 116X
44 10000 (12,120) 116X
45 10000 (12,120) 116X
46 10000 (12,120) 116X
47 10000 (12,120) 116X
48 10000 (12,120) 116X
49 10000 (12,120) 116X
50 10000 (12,120) 116X
51 10000 (12,120) 116X
52 10000 (12,120) 116X
53 10000 (12,120) 116X
54 10000 (12,120) 116X
55 10000 (12,120) 116X
56 10000 (12,120) 116X
57 10000 (12,120) 116X
58 10000 (12,120) 116X
59 10000 (12,120) 116X
60 10000 (12,120) 116X
61 10000 (12,120) 116X
62 10000 (12,120) 116X
63 10000 (12,120) 116X
64 10000 (12,120) 116X
65 10000 (12,120) 116X
66 10000 (12,120) 116X
67 10000 (12,120) 116X
68 10000 (12,120) 116X
69 10000 (12,120) 116X
70 10000 (12,120) 116X
71 10000 (12,120) 116X
72 10000 (12,120) 116X
73 10000 (12,120) 116X
74 10000 (12,120) 116X
75 10000 (12,120) 116X
76 10000 (12,120) 116X
77 10000 (12,120) 116X
78 10000 (12,120) 116X
79 10000 (12,120) 116X
80 10000 (12,120) 116X
81 10000 (12,120) 116X
82 10000 (12,120) 116X
83 10000 (12,120) 116X
84 10000 (12,120) 116X
85 10000 (12,120) 116X
86 10000 (12,120) 116X
87 10000 (12,120) 116X
88 10000 (12,120) 116X
89 10000 (12,120) 116X
90 10000 (12,120) 116X
91 10000 (12,120) 116X
92 10000 (12,120) 116X
93 10000 (12,120) 116X
94 10000 (12,120) 116X
95 10000 (12,120) 116X
96 10000 (12,120) 116X
97 10000 (12,120) 116X
98 10000 (12,120) 116X
99 10000 (12,120) 116X
100 10000 (12,120) 116X

```

```

1110 I=1,10
1120 J=1,10
1130 L=1,10
1140 M=1,10
1150 N=1,10
1160 O=1,10
1170 P=1,10
1180 Q=1,10
1190 R=1,10
1200 S=1,10
1210 T=1,10
1220 U=1,10
1230 V=1,10
1240 W=1,10
1250 X=1,10
1260 Y=1,10
1270 Z=1,10
1280 AA=1,10
1290 BB=1,10
1300 CC=1,10
1310 DD=1,10
1320 EE=1,10
1330 FF=1,10
1340 GG=1,10
1350 HH=1,10
1360 II=1,10
1370 JJ=1,10
1380 KK=1,10
1390 LL=1,10
1400 MM=1,10
1410 NN=1,10
1420 OO=1,10
1430 PP=1,10
1440 QQ=1,10
1450 RR=1,10
1460 SS=1,10
1470 TT=1,10
1480 UU=1,10
1490 VV=1,10
1500 WW=1,10
1510 XX=1,10
1520 YY=1,10
1530 ZZ=1,10
1540 AA=1,10
1550 BB=1,10
1560 CC=1,10
1570 DD=1,10
1580 EE=1,10
1590 FF=1,10
1600 GG=1,10
1610 HH=1,10
1620 II=1,10
1630 JJ=1,10
1640 KK=1,10
1650 LL=1,10
1660 MM=1,10
1670 NN=1,10
1680 OO=1,10
1690 PP=1,10
1700 QQ=1,10
1710 RR=1,10
1720 SS=1,10
1730 TT=1,10
1740 UU=1,10
1750 VV=1,10
1760 WW=1,10
1770 XX=1,10
1780 YY=1,10
1790 ZZ=1,10
1800 AA=1,10
1810 BB=1,10
1820 CC=1,10
1830 DD=1,10
1840 EE=1,10
1850 FF=1,10
1860 GG=1,10
1870 HH=1,10
1880 II=1,10
1890 JJ=1,10
1900 KK=1,10
1910 LL=1,10
1920 MM=1,10
1930 NN=1,10
1940 OO=1,10
1950 PP=1,10
1960 QQ=1,10
1970 RR=1,10
1980 SS=1,10
1990 TT=1,10
2000 UU=1,10
2010 VV=1,10
2020 WW=1,10
2030 XX=1,10
2040 YY=1,10
2050 ZZ=1,10
2060 AA=1,10
2070 BB=1,10
2080 CC=1,10
2090 DD=1,10
2100 EE=1,10
2110 FF=1,10
2120 GG=1,10
2130 HH=1,10
2140 II=1,10
2150 JJ=1,10
2160 KK=1,10
2170 LL=1,10
2180 MM=1,10
2190 NN=1,10
2200 OO=1,10
2210 PP=1,10
2220 QQ=1,10
2230 RR=1,10
2240 SS=1,10
2250 TT=1,10
2260 UU=1,10
2270 VV=1,10
2280 WW=1,10
2290 XX=1,10
2300 YY=1,10
2310 ZZ=1,10
2320 AA=1,10
2330 BB=1,10
2340 CC=1,10
2350 DD=1,10
2360 EE=1,10
2370 FF=1,10
2380 GG=1,10
2390 HH=1,10
2400 II=1,10
2410 JJ=1,10
2420 KK=1,10
2430 LL=1,10
2440 MM=1,10
2450 NN=1,10
2460 OO=1,10
2470 PP=1,10
2480 QQ=1,10
2490 RR=1,10
2500 SS=1,10
2510 TT=1,10
2520 UU=1,10
2530 VV=1,10
2540 WW=1,10
2550 XX=1,10
2560 YY=1,10
2570 ZZ=1,10
2580 AA=1,10
2590 BB=1,10
2600 CC=1,10
2610 DD=1,10
2620 EE=1,10
2630 FF=1,10
2640 GG=1,10
2650 HH=1,10
2660 II=1,10
2670 JJ=1,10
2680 KK=1,10
2690 LL=1,10
2700 MM=1,10
2710 NN=1,10
2720 OO=1,10
2730 PP=1,10
2740 QQ=1,10
2750 RR=1,10
2760 SS=1,10
2770 TT=1,10
2780 UU=1,10
2790 VV=1,10
2800 WW=1,10
2810 XX=1,10
2820 YY=1,10
2830 ZZ=1,10
2840 AA=1,10
2850 BB=1,10
2860 CC=1,10
2870 DD=1,10
2880 EE=1,10
2890 FF=1,10
2900 GG=1,10
2910 HH=1,10
2920 II=1,10
2930 JJ=1,10
2940 KK=1,10
2950 LL=1,10
2960 MM=1,10
2970 NN=1,10
2980 OO=1,10
2990 PP=1,10
3000 QQ=1,10
3010 RR=1,10
3020 SS=1,10
3030 TT=1,10
3040 UU=1,10
3050 VV=1,10
3060 WW=1,10
3070 XX=1,10
3080 YY=1,10
3090 ZZ=1,10
3100 AA=1,10
3110 BB=1,10
3120 CC=1,10
3130 DD=1,10
3140 EE=1,10
3150 FF=1,10
3160 GG=1,10
3170 HH=1,10
3180 II=1,10
3190 JJ=1,10
3200 KK=1,10
3210 LL=1,10
3220 MM=1,10
3230 NN=1,10
3240 OO=1,10
3250 PP=1,10
3260 QQ=1,10
3270 RR=1,10
3280 SS=1,10
3290 TT=1,10
3300 UU=1,10
3310 VV=1,10
3320 WW=1,10
3330 XX=1,10
3340 YY=1,10
3350 ZZ=1,10
3360 AA=1,10
3370 BB=1,10
3380 CC=1,10
3390 DD=1,10
3400 EE=1,10
3410 FF=1,10
3420 GG=1,10
3430 HH=1,10
3440 II=1,10
3450 JJ=1,10
3460 KK=1,10
3470 LL=1,10
3480 MM=1,10
3490 NN=1,10
3500 OO=1,10
3510 PP=1,10
3520 QQ=1,10
3530 RR=1,10
3540 SS=1,10
3550 TT=1,10
3560 UU=1,10
3570 VV=1,10
3580 WW=1,10
3590 XX=1,10
3600 YY=1,10
3610 ZZ=1,10
3620 AA=1,10
3630 BB=1,10
3640 CC=1,10
3650 DD=1,10
3660 EE=1,10
3670 FF=1,10
3680 GG=1,10
3690 HH=1,10
3700 II=1,10
3710 JJ=1,10
3720 KK=1,10
3730 LL=1,10
3740 MM=1,10
3750 NN=1,10
3760 OO=1,10
3770 PP=1,10
3780 QQ=1,10
3790 RR=1,10
3800 SS=1,10
3810 TT=1,10
3820 UU=1,10
3830 VV=1,10
3840 WW=1,10
3850 XX=1,10
3860 YY=1,10
3870 ZZ=1,10
3880 AA=1,10
3890 BB=1,10
3900 CC=1,10
3910 DD=1,10
3920 EE=1,10
3930 FF=1,10
3940 GG=1,10
3950 HH=1,10
3960 II=1,10
3970 JJ=1,10
3980 KK=1,10
3990 LL=1,10
4000 MM=1,10
4010 NN=1,10
4020 OO=1,10
4030 PP=1,10
4040 QQ=1,10
4050 RR=1,10
4060 SS=1,10
4070 TT=1,10
4080 UU=1,10
4090 VV=1,10
4100 WW=1,10
4110 XX=1,10
4120 YY=1,10
4130 ZZ=1,10
4140 AA=1,10
4150 BB=1,10
4160 CC=1,10
4170 DD=1,10
4180 EE=1,10
4190 FF=1,10
4200 GG=1,10
4210 HH=1,10
4220 II=1,10
4230 JJ=1,10
4240 KK=1,10
4250 LL=1,10
4260 MM=1,10
4270 NN=1,10
4280 OO=1,10
4290 PP=1,10
4300 QQ=1,10
4310 RR=1,10
4320 SS=1,10
4330 TT=1,10
4340 UU=1,10
4350 VV=1,10
4360 WW=1,10
4370 XX=1,10
4380 YY=1,10
4390 ZZ=1,10
4400 AA=1,10
4410 BB=1,10
4420 CC=1,10
4430 DD=1,10
4440 EE=1,10
4450 FF=1,10
4460 GG=1,10
4470 HH=1,10
4480 II=1,10
4490 JJ=1,10
4500 KK=1,10
4510 LL=1,10
4520 MM=1,10
4530 NN=1,10
4540 OO=1,10
4550 PP=1,10
4560 QQ=1,10
4570 RR=1,10
4580 SS=1,10
4590 TT=1,10
4600 UU=1,10
4610 VV=1,10
4620 WW=1,10
4630 XX=1,10
4640 YY=1,10
4650 ZZ=1,10
4660 AA=1,10
4670 BB=1,10
4680 CC=1,10
4690 DD=1,10
4700 EE=1,10
4710 FF=1,10
4720 GG=1,10
4730 HH=1,10
4740 II=1,10
4750 JJ=1,10
4760 KK=1,10
4770 LL=1,10
4780 MM=1,10
4790 NN=1,10
4800 OO=1,10
4810 PP=1,10
4820 QQ=1,10
4830 RR=1,10
4840 SS=1,10
4850 TT=1,10
4860 UU=1,10
4870 VV=1,10
4880 WW=1,10
4890 XX=1,10
4900 YY=1,10
4910 ZZ=1,10
4920 AA=1,10
4930 BB=1,10
4940 CC=1,10
4950 DD=1,10
4960 EE=1,10
4970 FF=1,10
4980 GG=1,10
4990 HH=1,10
5000 II=1,10
5010 JJ=1,10
5020 KK=1,10
5030 LL=1,10
5040 MM=1,10
5050 NN=1,10
5060 OO=1,10
5070 PP=1,10
5080 QQ=1,10
5090 RR=1,10
5100 SS=1,10
5110 TT=1,10
5120 UU=1,10
5130 VV=1,10
5140 WW=1,10
5150 XX=1,10
5160 YY=1,10
5170 ZZ=1,10
5180 AA=1,10
5190 BB=1,10
5200 CC=1,10
5210 DD=1,10
5220 EE=1,10
5230 FF=1,10
5240 GG=1,10
5250 HH=1,10
5260 II=1,10
5270 JJ=1,10
5280 KK=1,10
5290 LL=1,10
5300 MM=1,10
5310 NN=1,10
5320 OO=1,10
5330 PP=1,10
5340 QQ=1,10
5350 RR=1,10
5360 SS=1,10
5370 TT=1,10
5380 UU=1,10
5390 VV=1,10
5400 WW=1,10
5410 XX=1,10
5420 YY=1,10
5430 ZZ=1,10
5440 AA=1,10
5450 BB=1,10
5460 CC=1,10
5470 DD=1,10
5480 EE=1,10
5490 FF=1,10
5500 GG=1,10
5510 HH=1,10
5520 II=1,10
5530 JJ=1,10
5540 KK=1,10
5550 LL=1,10
5560 MM=1,10
5570 NN=1,10
5580 OO=1,10
5590 PP=1,10
5600 QQ=1,10
5610 RR=1,10
5620 SS=1,10
5630 TT=1,10
5640 UU=1,10
5650 VV=1,10
5660 WW=1,10
5670 XX=1,10
5680 YY=1,10
5690 ZZ=1,10
5700 AA=1,10
5710 BB=1,10
5720 CC=1,10
5730 DD=1,10
5740 EE=1,10
5750 FF=1,10
5760 GG=1,10
5770 HH=1,10
5780 II=1,10
5790 JJ=1,10
5800 KK=1,10
5810 LL=1,10
5820 MM=1,10
5830 NN=1,10
5840 OO=1,10
5850 PP=1,10
5860 QQ=1,10
5870 RR=1,10
5880 SS=1,10
5890 TT=1,10
5900 UU=1,10
5910 VV=1,10
5920 WW=1,10
5930 XX=1,10
5940 YY=1,10
5950 ZZ=1,10
5960 AA=1,10
5970 BB=1,10
5980 CC=1,10
5990 DD=1,10
6000 EE=1,10
6010 FF=1,10
6020 GG=1,10
6030 HH=1,10
6040 II=1,10
6050 JJ=1,10
6060 KK=1,10
6070 LL=1,10
6080 MM=1,10
6090 NN=1,10
6100 OO=1,10
6110 PP=1,10
6120 QQ=1,10
6130 RR=1,10
6140 SS=1,10
6150 TT=1,10
6160 UU=1,10
6170 VV=1,10
6180 WW=1,10
6190 XX=1,10
6200 YY=1,10
6210 ZZ=1,10
6220 AA=1,10
6230 BB=1,10
6240 CC=1,10
6250 DD=1,10
6260 EE=1,10
6270 FF=1,10
6280 GG=1,10
6290 HH=1,10
6300 II=1,10
6310 JJ=1,10
6320 KK=1,10
6330 LL=1,10
6340 MM=1,10
6350 NN=1,10
6360 OO=1,10
6370 PP=1,10
6380 QQ=1,10
6390 RR=1,10
6400 SS=1,10
6410 TT=1,10
6420 UU=1,10
6430 VV=1,10
6440 WW=1,10
6450 XX=1,10
6460 YY=1,10
6470 ZZ=1,10
6480 AA=1,10
6490 BB=1,10
6500 CC=1,10
6510 DD=1,10
6520 EE=1,10
6530 FF=1,10
6540 GG=1,10
6550 HH=1,10
6560 II=1,10
6570 JJ=1,10
6580 KK=1,10
6590 LL=1,10
6600 MM=1,10
6610 NN=1,10
6620 OO=1,10
6630 PP=1,10
6640 QQ=1,10
6650 RR=1,10
6660 SS=1,10
6670 TT=1,10
6680 UU=1,10
6690 VV=1,10
6700 WW=1,10
6710 XX=1,10
6720 YY=1,10
6730 ZZ=1,10
6740 AA=1,10
6750 BB=1,10
6760 CC=1,10
6770 DD=1,10
6780 EE=1,10
6790 FF=1,10
6800 GG=1,10
6810 HH=1,10
6820 II=1,10
6830 JJ=1,10
6840 KK=1,10
6850 LL=1,10
6860 MM=1,10
6870 NN=1,10
6880 OO=1,10
6890 PP=1,10
6900 QQ=1,10
6910 RR=1,10
6920 SS=1,10
6930 TT=1,10
6940 UU=1,10
6950 VV=1,10
6960 WW=1,10
6970 XX=1,10
6980 YY=1,10
6990 ZZ=1,10
7000 AA=1,10
7010 BB=1,10
7020 CC=1,10
7030 DD=1,10
7040 EE=1,10
7050 FF=1,10
7060 GG=1,10
7070 HH=1,10
7080 II=1,10
7090 JJ=1,10
7100 KK=1,10
7110 LL=1,10
7120 MM=1,10
7130 NN=1,10
7140 OO=1,10
7150 PP=1,10
7160 QQ=1,10
7170 RR=1,10
7180 SS=1,10
7190 TT=1,10
7200 UU=1,10
7210 VV=1,10
7220 WW=1,10
7230 XX=1,10
7240 YY=1,10
7250 ZZ=1,10
7260 AA=1,10
7270 BB=1,10
7280 CC=1,10
7290 DD=1,10
7300 EE=1,10
7310 FF=1,10
7320 GG=1,10
7330 HH=1,10
7340 II=1,10
7350 JJ=1,10
7360 KK=1,10
7370 LL=1,10
7380 MM=1,10
7390 NN=1,10
7400 OO=1,10
7410 PP=1,10
7420 QQ=1,10
7430 RR=1,10
7440 SS=1,10
7450 TT=1,10
7460 UU=1,10
7470 VV=1,10
7480 WW=1,10
7490 XX=1,10
7500 YY=1,10
7510 ZZ=1,10
7520 AA=1,10
7530 BB=1,10
7540 CC=1,10
7550 DD=1,10
7560 EE=1,10
7570 FF=1,10
7580 GG=1,10
7590 HH=1,10
7600 II=1,10
7610 JJ=1,10
7620 KK=1,10
7630 LL=1,10
7640 MM=1,10
7650 NN=1,10
7660 OO=1,10
7670 PP=1,10
7680 QQ=1,10
7690 RR=1,10
7700 SS=1,10
7710 TT=1,10
7720 UU=1,10
7730 VV=1,10
7740 WW=1,10
7750 XX=1,10
7760 YY=1,10
7770 ZZ=1,10
7780 AA=1,10
7790 BB=1,10
7800 CC=1,10
7810 DD=1,10
7820 EE=1,10
7830 FF=1,10
7840 GG=1,10
7850 HH=1,10
7860 II=1,10
7870 JJ=1,10
7880 KK=1,10
7890 LL=1,10
7900 MM=1,10
7910 NN=1,10
7920 OO=1,10
7930 PP=1,10
7940 QQ=1,10
7950 RR=1,10
7960 SS=1,10
7970 TT=1,10
7980 UU=1,10
7990 VV=1,10
8000 WW=1,10
8010 XX=1,10
8020 YY=1,10
8030 ZZ=1,10
8040 AA=1,10
8050 BB=1,10
8060 CC=1,10
8070 DD=1,10
8080 EE=1,10
8090 FF=1,10
8100 GG=1,10
8110 HH=1,10
8120 II=1,10
8130 JJ=1,10
8140 KK=1,10
8150 LL=1,10
8160 MM=1,10
8170 NN=1,10
8180 OO=1,10
8190 PP=1,10
8200 QQ=1,10
8210 RR=1,10
8220 SS=1,10
8230 TT=1,10
8240 UU=1,10
8250 VV=1,10
8260 WW=1,10
8270 XX=1,10
8280 YY=1,10
8290 ZZ=1,10
8300 AA=1,10
8310 BB=1,10
8320 CC=1,10
8330 DD=1,10
8340 EE=1,10
8350 FF=1,10
8360 GG=1,10
8370 HH=1,10
8380 II=1,10
8390 JJ=1,10
8400 KK=1,10
8410 LL=1,10
8420 MM=1,10
8430 NN=1,10
8440 OO=1,10
8450 PP=1,10
8460 QQ=1,10
8470 RR=1,10
8480 SS=1,10
8490 TT=1,10
8500 UU=1,10
8510 VV=1,10
8520 WW=1,10
8530 XX=1,10
8540 YY=1,10
8550 ZZ=1,10
8560 AA=1,10
8570 BB=1,10
8580 CC=1,10
8590 DD=1,10
8600 EE=1,10
8610 FF=1,10
8620 GG=1,10
8630 HH=1,10
8640 II=1,10
8650 JJ=1,10
8660 KK=1,10
8670 LL=1,10
8680 MM=1,10
8690 NN=1,10
8700 OO=1,10
8710 PP=1,10
8720 QQ=1,10
8730 RR=1,10
8740 SS=1,10
8750 TT=1,10
8760 UU=1,10
8770 VV=1,10
8780 WW=1,10
8790 XX=1,10
8800 YY=1,10
8810 ZZ=1,10
8820 AA=1,10
8830 BB=1,10
8840 CC=1,10
8850 DD=1,10
8860 EE=1,10
8870 FF=1,10
8880 GG=1,10
8890 HH=1,10
8900 II=1,10
8910 JJ=1,10
8920 KK=1,10
8930 LL=1,10
8940 MM=1,10
8950 NN=1,10
8960 OO=1,10
8970 PP=1,10
8980 QQ=1,10
8990 RR=1,10
9000 SS=1,10
9010 TT=1,10
9020 UU=1,10
9030 VV=1,10
9040 WW=1,10
9050 XX=1,10
9060 YY=1,10
9070 ZZ=1,10
9080 AA=1,10
9090 BB=1,10
9100 CC=1,10
9110 DD=1,10
9120 EE=1,10
9130 FF=1,10
9140 GG=1,10
9150 HH=1,10
9160 II=1,10
9170 JJ=1,10
9180 KK=1,10
9190 LL=1,10
9200 MM=1,10
9210 NN=1,10
9220 OO=1,10
9230 PP=1,10
9240 QQ=1,10
9250 RR=1,10
9260 SS=1,10
9270 TT=1,10
9280 UU=1,10
9290 VV=1,10
9300 WW=1,10
9310 XX=1,10
9320 YY=1,10
9330 ZZ=1,10
9340 AA=1,10
9350 BB=1,10
9360 CC=1,10
9370 DD=1,10
9380 EE=1,10
9390 FF=1,10
9400 GG=1,10
9410 HH=1,10
9420 II=1,10
9430 JJ=1,10
9440 KK=1,10
9450 LL=1,10
9460 MM=1,10
9470 NN=1,10
9480 OO=1,10
9490 PP=1,10
9500 QQ=1,10
9510 RR=1,10
9520 SS=1,10
9530 TT=1,10
9540 UU=1,10
9550 VV=1,10
9560 WW=1,10
9570 XX=1,10
9580 YY=1,10
9590 ZZ=1,10
9600 AA=1,10
9610 BB=1,10
9620 CC=1,10
9630 DD=1,10
9640 EE=1,10
9650 FF=1,10
9660 GG=1,10
9670 HH=1,10
9680 II=1,10
9690 JJ=1,10
9700 KK=1,10
9710 LL=1,10
9720 MM=1,10
9730 NN=1,10
9740 OO=1,10
9750 PP=1,10
9760 QQ=1,10
9770 RR=1,10
9780 SS=1,10
9790 TT=1,10
9800 UU=1,10
9810 VV=1,10
9820 WW=1,10
9830 XX=1,10
9840 YY=1,10
9850 ZZ=1,10
9860 AA=1,10
9870 BB=1,10
9880 CC=1,10
9890 DD=1,10
9900 EE=1,10
9910 FF=1,10
9920 GG=1,10
9930 HH=1,10
9940 II=1,10
9950 JJ=1,10
9960 KK=1,10
9970 LL=1,10
9980 MM=1,10
9990 NN=1,10

```


APPENDIX B

PROGRAM LISTING

FOR A

SEQUENTIAL FIRST-ORDER RESPONSE SURFACE METHOD

```

C
      DIMENSION XVAR(30),SID(30),RY(30),
      YV(30),Sb(30),IV(30),RX(30,30),RI(65),ANS(10),
      XP(10),XL(10),
      C CLAMIN X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),
      YI(10),GRJ(20),N,AVTEX,NV,M,MOB,GR,STEP,RAD,
      REFA,CONF,CLEU,ISLUP,NIV
      C CLAMIN/BLSEEL/ISLEO1,ISLEO2
      C CLAMIN/BLSEEL/AL(10),XU(10),YL,YU
      C CLAMIN/BL4/JACI(30)
      C CLAMIN/BL5/JV(30),ISAVE(30)
      C CLAMIN/BLSE3/ITER
      WRITE (6,2000)
      READ (9,*) ZOSIM
      READ (9,*) N,NK,MCP
      READ (9,*) (XL(J),J=1,N), (XU(J),J=1,N)
      READ (9,*) (XLP(I),I=1,N), (XUP(I),I=1,N)
      READ (9,*) (DS(J),J=1,9)
      READ (9,*) YL,YU
      READ (9,*) STEP,RAD,KRKA
      READ (9,*) CONF
      READ (9,*) IX
      READ (9,*) ISEED1,ISLEO2
      C
      WRITE (12,*) MDSIM
      WRITE (12,*) N,NK,MCP
      WRITE (12,*) (XL(J),J=1,N), (XU(J),J=1,N)
      WRITE (12,*) (XLP(I),I=1,N), (XUP(I),I=1,N)
      WRITE (12,*) YL,YU
      WRITE (12,*) STEP,RAD,KRKA
      WRITE (12,*) CONF
      WRITE (12,*) IX
      WRITE (12,*) ISEED1,ISLEO2
      C
      SOL=(-1.+SQRT(5.))/2.
      YEX=0
      DO 10 I=10,20, MUDEV
      WRITE (6,2015)
      AVTEX=N+1
      DO 20 I=10,20
      C
      WRITE (6,2025)
      AVTEX=2*N
      C
      AVTEX=AVTEX
      R=AVTEX+1
      MPI=4+I
      MOB=AVTEX+MCP
      NV=N+1
      NIV=N+NK
      WRITE (6,2035)
      WRITE (6,2040)
      DO 45 J=1,NIV
      ISAVE(J)=J
      CONTINUE
      DO 50 J=1,N
      CALL RANDU(IX,IY,RJ,N)

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590

```

```

1X=IY
JA=XU(J)-XL(J)
X(H,J)=XL(J)+RJMN*DA
XC(J)=X(H,J)
60 CONTINUE
C
CALL FSUPT(IOUT,JACT,XC)
IF (IOUT.EQ.1) GO TO 50
DO 90 K=1,NR
J=4+K
X(M,J)=Y(K)
XC(J)=Y(K)
CONTINUE
1000 IF (MCP.EQ.1) GO TO 1050
DO 90 I=1,NPI,MUB
IF (MOSIA.EQ.1) GO TO 83
CALL SIMUL(XC,Y)
DO 85 K=1,NR
J=4+K
X(I,J)=Y(K)
CONTINUE
85 CONTINUE
90 CONTINUE
1050 WRITE (6,2050) ITER
1060 WRITE (6,2060) IFLK,MCP
95 GO TO (100,150), MDEV
C
100 CALL SIMPLX(XC)
DO 1100
CALL FACTOR(XC)
1100 WRITE (6,2100) ITER,MVTEX
DO 350 J=1,NV
DO 350 I=1,MUB
K=MUB*(J-1)+I
XV(K)=X(I,J)
CONTINUE
350 CALL CURR(MUB,NV,I,XV,XJAK,STD,RX,RT,BV,DV,TV)
CALL CRER(NV,KI,NV,L,ISAVE,RX,RY)
CALL RINV(RX,N,DET,DV,TV)
WRITE (6,320) (X(I,J),J=1,NTV),I=1,MUB)
320 EODMAT(1/(5X,4F1+.6))
CALL VULTR(4UB,N,XJAK,STD,BV,RX,RY,ISAVE,DV,SR,TV,ANS)
WRITE (6,2150) IFLK,XC(J),J=1,N), (DV(J),J=1,N)
WRITE (6,2200) (DV(J),J=1,N)
IF (IOUT.EQ.1) GO TO 580
CALL PROFCT(JACT)
WRITE (6,2300) (DV(J),J=1,N)
380 SUMSQ=0
DO 400 J=1,N
SUMSQ=SUMSQ+DV(J)*DV(J)
400 CONTINUE
DO 450 J=1,N
280 (J)=DV(J)/SQRT(SUMSQ)
CALL SUDEN(IOUT,JACT,XC)
IF (ISUDEN.EQ.1) GO TO 500
ITER=ITER+1
90 GO TO 1000

```



```

      X2=1.00 XK1=1.2
      DO 300 KK2=1,2
      I=2*(KK1-1)+KK2
      X(I,1)=XC(1)+RAD*K1
      X(I,2)=XC(2)+RAD*K2
      DO 100 J=1,N
      XP(J)=X(I,J)
      XC(I)=XP(J)
      80 CONTINUE
      CALL SIMUL(XP,Y)
      DO 100 K=1,NR
      J=N+K
      X(I,J)=Y(K)
      XP(J)=Y(K)
      XC(I)=XP(J)
      100 CONTINUE
      K2=-K2
      XC(I)=X1
      200 CONTINUE
      K1=-K1
      XC(I)=XP(J)
      300 CONTINUE
      RAD=RAD*ERRA
      RETURN
      END

      C
      SUBROUTINE GOLDEN(IOUT,JACT,XC)
      DIMENSION XGL(10),XGU(10),XG1(10),XG2(10),XP(10),XC(10),KACT(20),
      JACT(30)
      COMMON X(40,30),XV(1200),XL(20),XU(20),YL(10),YU(10),
      YI(10),GRAD(120),N,KVIEK,NV,M,MUB,NK,STEP,RAD,
      PRRA,CONF,GLL,ISUP,NTV
      ISTOP=0
      IOUT=1
      Y1=MAX=XC(NV)
      DO 50 J=1,NTV
      XGL(J)=XC(J)
      XC(J)=XGL(J)
      50 CONTINUE
      ISTOP=STEP
      100 XP(J)=XC(J)+I*STEP*GRAD(J)
      150 CONTINUE
      CALL FSPRT(JOUT,JACT,XP)
      JOUT=IOUT+JOUT
      IF(JOUT.EQ.1) GO TO 950
      DO 100 (200,240,200),IND
      J=1,NTV
      XGL(J)=XC(J)
      200 CONTINUE
      220 XGL(J)=XGL(J)
      240 XGL(J)=XGL(J)
      260 XGL(J)=XGL(J)
      280 XGL(J)=XGL(J)
      300 XGL(J)=XGL(J)
      J=1,NTV
      XGL(J)=Y(K)
      320 CONTINUE
      IND=IND+1
      IF(XY(1).LE.Y1MAX) GO TO 400

```

```

IF (ICUT.NE.0) GO TO 430
YMAX=Y(I)
STEP=STEP/GOLD
ISTEP=ISTEP+STEP
GO TO 100
C 400 IF (IND.NE.2) GO TO 490
430 GO 450 J=1,N
XG1(J)=XGL(J)+STEP*GOLD*GRAD(J)
450 CONTINUE
CALL SIMUL(XG1,Y)
DO 490 K=1,NR
J=N+K
XG1(J)=Y(K)
430 CONTINUE
GO TO 500
490 STEP=STEP/GOLD
500 DO 550 J=1,N
XG2(J)=XGL(J)+STEP*GOLD*GRAD(J)
550 CONTINUE
CALL SIMUL(XG2,Y)
DO 590 K=1,NR
J=N+K
XG2(J)=Y(K)
580 CONTINUE
C 600 STEP=STEP*GOLD
IF (XG1(NV).LT.XG2(NV)) GO TO 700
DO 650 J=1,NV
XG2(J)=XGL(J)
XG1(J)=XGL(J)+STEP*GOLD*GOLD*GRAD(J)
650 CONTINUE
CALL SIMUL(XG1,Y)
DO 690 K=1,NR
J=N+K
XG1(J)=Y(K)
630 CONTINUE
GO TO 800
700 DO 750 J=1,NV
XGL(J)=XGL(J)
XG1(J)=XG2(J)
XG2(J)=XGL(J)+STEP*GOLD*GRAD(J)
750 CONTINUE
CALL SIMUL(XG2,Y)
DO 790 K=1,NR
J=N+K
XG2(J)=Y(K)
780 CONTINUE
C 800 IF (STEP.LT.CONF) GO TO 950
GO TO 600
850 STEP=STEP*GOLD
IF (IND.NE.1) GO TO 900
IF (STEP.LT.CONF) GO TO 1200
GO TO 800
900 IF (IND.NE.2) GO TO 490
GO TO 1040
950 IF (XG1(NV).LT.XG2(NV)) GO TO 1050
DO 1000 J=1,NV

```

```

00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

```

```

1000 XC(J)=XG1(J)
1001 X(J)=XC(J)
1002 CONTINUE
1050 GO TO 1300
1051 J=1,NIV
1052 XC(J)=XG2(J)
1053 X(J)=XC(J)
1054 CONTINUE
1100 GO TO 1300
1101 ISTOP=1
1200 RETURN
1300 END

C
SUBROUTINE FSRPT(IOUT,JACT,XT)
DIMENSION XT(2),JACT(30)
COMMON X(40),XV(120),AL(20),XU(20),YL(10),YU(10),RAD,
& RARA,CONF,GULO,ISTOP,NIV
COMMON/BLEEF/XL(10),XUP(10)
IOUT=0
CALL SIMUL(AT,Y,NR)
GO TO 50 J=1,NIV
JACT(J)=0
CONTINUE
50 IF (XT(J)).GE.XLP(J) GO TO 120
GO TO 140
120 IF (XT(J)).LE.XUP(J) GO TO 160
140 JACT(J)=1
160 CONTINUE
GO TO 260 L=2,NR
J=N+L
IF (Y(L).GE.YL) GO TO 220
GO TO 240
220 IF (Y(L).LE.YU) GO TO 260
240 JACT(J)=1
260 CONTINUE
RETURN
END

C
SUBROUTINE PRODT(JACT)
DIMENSION G(4,4),G1RA(4,4),GPRJ(4,4),GPRJ1(4,4),GV(10),
& DV1(30),DV2(30),RX(30),RX(30),RI(455),
& DRY(30),SV(30),AS(10),JACT(30)
COMMON X(40),XV(120),AL(20),XU(20),YL(10),YU(10),
& RARA,CONF,GULO,ISTOP,NIV
COMMON/BLEEF/XL(10),XUP(10)
IOUT=0
GO TO 100 J=1,NIV
JACT(J)=0
CONTINUE
100 IF (JACT(J)).GE.1
GO TO 120
120 K=1,N
GO TO 140 K=1,N
140 CONTINUE
160 G(IOUTS,J)=1.

```

```

0002960
0002970
0002980
0002990
0003000
0003010
0003020
0003030
0003040
0003050
0003060
0003070
0003080
0003090
0003100
0003110
0003120
0003130
0003140
0003150
0003160
0003170
0003180
0003190
0003200
0003210
0003220
0003230
0003240
0003250
0003260
0003270
0003280
0003290
0003300
0003310
0003320
0003330
0003340
0003350
0003360
0003370
0003380
0003390
0003400
0003410
0003420
0003430
0003440
0003450
0003460
0003470
0003480
0003490
0003500
0003510
0003520
0003530
0003540

```


APPENDIX C

PROGRAM LISTING

FOR A

SECOND-ORDER RESPONSE SURFACE METHOD

BASED ON A CENTRAL COMPOSITE DESIGN


```

120 DO 220 I=1,NP
140 CALL RANDU(IX,IY,RUAN)
IX=IY
XP(J)=XLP(J)+RUMN*DA(J)
X(I,J)=XP(J)
140 CONTINUE
CALL SECIZE(XP)
CALL FSBPI(XP,Y,IOUT)
IF (ICUT.EC.2) GO TO 120
ITER2=ITER2+1
WRITE (12,1450) ITER2
CALL WRITE(XP,Y,N,GA,ICUT)
IF (ICUT.EC.0) GO TO 120
DO 200 J=1,NR
YP(I,J)=Y(J)
200 CONTINUE
YI(IJ)=Y(IJ)
IF (YI(IJ).GT.YI(MAX)) MAX=I
220 CONTINUE
WRITE (12,1230) NP
C
300 CALL WORST(YI,MIN)
DO 340 J=1,N
SUM=X(MIN,J)
DO 320 I=1,NP
SUM=SUM+X(I,J)
320 CONTINUE
C(J)=SUM/ANPML
340 CONTINUE
C
R=2*
R=70.5
IF (R.LT.1.E-2) GO TO 600
DO 420 J=1,N
XP(J)=C(J)+R*(C(J)-X(MIN,J))
420 CONTINUE
CALL SECIZE(XP)
CALL FSBPI(XP,Y,IOUT)
IF (ICUT.EC.2) GO TO 400
ITER2=ITER2+1
WRITE (12,1450) ITER2
CALL WRITE(XP,Y,N,NR,IOUT)
IF (ICUT.EC.0) GO TO 400
IF (YI(IJ).LE.YI(MIN)) GO TO 400
DO 440 J=1,NR
YP(MIN,J)=Y(J)
440 CONTINUE
DO 500 J=1,N
X(MIN,J)=XP(J)
500 CONTINUE
YI(MIN)=Y(IJ)
IF (YI(MIN).GT.YI(MAX)) MAX=MIN
GO TO 300
C
600 WRITE (12,1600)
DO 620 J=1,N
WRITE (12,1160) J,X(MAX,J)
620 CONTINUE

```

```

00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130
00001140
00001150
00001160
00001170
00001180

```

```

DO 640 J=1,NR
WRITE (12,1180) J,YP(MAX,J)
CONTINUE
640
C
DO 700 J=1,N
XP(J)=X(MAX,J)
CONTINUE
700
CALL AVSSIM(XP,Y,N,NR)
WRITE (12,1600)
CALL WRITE(XP,Y,N,NR,0)
C
1050 FORMAT (1H1/35X,'***** LIST OF INPUT DATA *****',//)
1100 FORMAT (30X,'X(1,1) = ',F14.6)
1180 FORMAT (30X,'Y(1,1) = ',F14.6)
1230 FORMAT (7/5X,'THE ABOVE',12,' POINTS ARE THE INITIAL POINTS')
1450 FORMAT (//1X,13,' 1H ITERATION',//)
1600 FORMAT (//5X,'OPTIMAL SOLUTION',//)
5000 STOP
END
C
SUBROUTINE CENCOM
C***** THIS SUBROUTINE GENERATES THE EXPERIMENT POINTS FOR THE
C***** CENTRAL COMPOSITE DESIGN
C
DIMENSION Y(20),ED(10),XP(30)
CALL YN,N,AN,AVI,N20,N2DPI,NR,M,MCP,MOS,MDSIM,NP
CALL XN,BL5/RAC(10),XC(10)
CALL YN/BL7/AL(1000),YE(1000)
CALL XN/BL9/EX(1000),EY(1000)
C
SORTN=SQRT(AN)
MFAC=2*#N
MAXIS=2*#N
MVTEX=MFAC+MAXIS
I=AVTEX+1
MOS=AVTEX+MCP
VAL=N-1
N2D=N*(N+3)/2
N2DPI=N2D+1
C
DO 50 J=1,N
ED(J)=RAD(J)/SQRTN
CONTINUE
50
C
KX=0
KY=0
DO 160 K=1,MFAC
KA=K
DO 120 J=1,N
KB=JOD(KA,2)
XA=KA/2
XP(J)=XC(J)+ED(J)*(-1)**KB
CONTINUE
120
CALL SIZE(XP)
CALL AVSSIM(XP,Y,N,NR)
CALL RECORD(XP,Y,KX,KY)
CONTINUE
160

```

```

C
DO 220 I=1,N
DO 240 JA=1,2
DO 220 J=1,N
XP(J)=XC(J)
IF (I.EQ.J) XP(J)=XC(J)+RAD(J)*(-1)**IA
220 CONTINUE
CALL SECIZE(XP)
CALL AVGSIM(XP,Y,N,NR)
CALL RECORD(XP,Y,KX,KY)
240 CONTINUE
280 CONTINUE
C
CALL SECIZE(XC)
CALL AVGSIM(XC,Y,N,NR)
CALL RECORD(XC,Y,KX,KY)
C
IF (MCP.EQ.1) GO TO 500
DO 400 I=2,MCP
IF (MDSIM.EQ.1) GO TO 300
CALL AVGSIM(XC,Y,N,NR)
CALL RECORD(XC,Y,KX,KY)
300 CONTINUE
400 CONTINUE
C
WRITE (12,1450)
DO 500 I=1,MCP
KX1=(I-1)*N2D+1
KY1=(I-1)*NR+1
KY2=I*NK
WRITE (12,1600) I,(EX(J),J=KX1,KX2),(EY(J),J=KY1,KY2)
500 CONTINUE
C
CALL GMTRA(EX,XE,N2D,MOB)
CALL GMTRA(EY,YE,NR,MCS)
C
C1450 FORMAT (///50X,'REGRESSION TABLEAU',//12X,'X1',12X,'X2',11X,'X3',12X,'X4',/
&10X,'X5',12X,'X6',/
&12X,'Y5',12X,'Y6',/
1600 FORMAT(/1X,13,3X,9(F12.6,2X)/7X,9(F12.6,2X)/)
RETURN
END
C
SUBROUTINE SECIZE(XS)
C
C***** THIS SUBROUTINE TRANSFORMS 1ST ORDER INDEPENDENT VARIABLES
C***** INTO 2ND ORDER INDEPENDENT VARIABLES FOR
C***** 2ND ORDER REGRESSION
C
DIMENSION XS(30)
COMMON N,AN,N*1,N2D,N2D*1,NR,M,MCP,MOB,MDSIM,NP
K=N
DO 100 J=1,N
K=K+1
XS(K)=XS(J)*XS(J)
100 CONTINUE
C
DO 300 I=1,NM1
IP1=I+1

```



```

00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

DC 200 J=IP1,N
K=X+1
XS(K)=XS(1)*XS(J)
200 CONTINUE
300 RETURN
END

C
C SUBROUTINE REGRES
C ***** THIS SUBROUTINE CALCULATES THE REGRESSION COEFFICIENTS OF THE
C ***** RESPONSE SURFACES
C
C DIMENSION XBAR(30),STD(30),RX(900),RT(465),BV(30),DV(30),TV(30),
C          ISAVE(30),RY(30),SB(30),ANS(10)
C COMMON N,AN,N1,N2,N2DP1,NR,M,MCF,MOR,MDSIM,NP
C COMON/BL6/BL(10,30)
C 1117/BL7/XE(1000),YE(1000)
C
C NXE= 408*N2D
C
C 70 50 J=1,N2DP1
C      ISAVE(J)=J
C 50 CONTINUE
C
C      WRITE (12,1150)
C 70 300 J=1,NR
C      XSE=XE
C      KYE=MOB*(J-1)
C 100 100 I=1,NDB
C      XSE=XSE+1
C      KYE=KYE+1
C      XE(KXF)=YE(KYE)
C 100 CONTINUE
C
C      CALL CORPE(XCB,N2DP1,1,XE,XBAR,STD,RX,RT,BV,DV,TV)
C      CALL ORDER(N2DP1,RT,N2DP1,N2D,1,SAVE,RX,RY)
C      CALL MINV(RX,N2D,DEL,DV,TV)
C      CALL MULT(R(408),N2D,XBAR,STD,BV,RX,RY,ISAVE,DV,SB,TV,ANS)
C      WRITE (12,1160)((DV(L),L=1,N2D),ANS(1),J)
C      WRITE (12,1170) ANS(10)
C
C 200 JCOEFF=L,N2D
C      B(J,JCOEFF)=DV(JCOEFF)
C 200 CONTINUE
C 300 J(N2DP1)=ANS(1)
C 300 CONTINUE
C
C 1150 FORMAT (///1X,'THE ESTIMATED RESPONSE SURFACES ARE :')
C 1160 5X,F12.6,' X(1) + ',F12.6,' X(2) + ',F12.6,' X(1)*X(2) + ',
C      5X,F12.6,' Y(1),',
C 1170 FORMAT (//5X,'F VALUE = ',F14.6)
C      RETURN
C      EN
C
C SUBROUTINE PSDPT(XI,Y,IOUT)
C

```

```

C***** THIS SUBROUTINE CALCULATES THE RESPONSES AND
C***** DEFICITS IF THEY ARE FEASIBLE
C
      DIMENSION XT(30),Y(20)
      COMMON/BL6E3/XL(10),N2DPL,NR,M,MCP,MOB,MDSIM,NP
      COMMON/BL6E1/XLP(10),YL,YU
      COMMON/BL6E1/XLP(10),XUP(10)
      IOUT=2
      DO 100 J=1,N
        IF (XT(J)-LT,XLP(J)) GO TO 500
        IF (XT(J)-GT,XUP(J)) GO TO 500
        CONTINUE
      100 IOUT=1
      DO 300 I=2,NR
        Y(I)=B(1,N2DPL)
        DO 200 J=1,N2D
          Y(I)=Y(I)+B(I,J)*XT(J)
        CONTINUE
      200 IF (Y(I)-LT,YL) GO TO 500
        IF (Y(I)-GT,YU) GO TO 500
      300 CONTINUE
        Y(I)=B(1,N2DPL)
        DO 350 J=1,N2D
          Y(I)=Y(I)+B(I,J)*XT(J)
        CONTINUE
      350 IOUT=J
      DO 500 RETURN
      EN
C
      SUBROUTINE WORST(YT,MIN)
C***** THIS SUBROUTINE FINDS OUT THE WORST POINT IN THE
C***** POLYGON
C
      DIMENSION YT(30)
      COMMON/BL6E1/XL(10),N2DPL,NR,M,MCP,MOB,MDSIM,NP
      MIN=1
      YMIN=YT(1)
      DO 50 J=2,NP
        IF (YT(J)-GE,YMIN) GO TO 50
        MIN=J
        YMIN=YT(J)
      CONTINUE
      50 RETURN
      END
C
      SUBROUTINE DATA(NV,JV)
C***** THIS SUBROUTINE IS CALLED BY THE SSP SUBROUTINE 'CORRE'
C***** TO READ IN THE EXPERIMENT DATA NEEDED FOR REGRESSION
C
      DIMENSION DV(1)
      1 FORMAT (12F6.0)
      1 READ (3,1) (DV(J),J=1,NV)

```

```

WRITE (1) (OV(J),J=1,NV)
RETURN
END
C
SUBROUTINE WRITE(XW,YW,N,NR,ICODE)
DIMENSION XW(20),YW(20)
C
IF (ICODE.NE.J) GO TO 500
C
DO 120 J=1,N
WRITE (12,1100) J,XW(J)
CONTINUE
DO 200 J=1,NR
WRITE (12,1200) J,YW(J)
CONTINUE
GO TO 1000
C
DO 500 J=1,N
WRITE (12,1200) J,XW(J)
CONTINUE
DO 700 J=1,NR
WRITE (12,1700) J,YW(J)
CONTINUE
C
1100 FORMAT (30X,'X(',11,') = ',F14.6)
1200 FORMAT (30X,'Y(',11,') = ',F14.6)
1600 FORMAT (80X,'X(',11,') = ',F14.6)
1700 FORMAT (80X,'Y(',11,') = ',F14.6)
1000 RETURN
END
C
SUBROUTINE RECORD(XP,Y,KX,KY)
DIMENSION XP(30),Y(20)
COMMON N,AN,NV1,N20,N20PL,NR,M,MCP,MCB,MDSIM,NP
COMMON/BL9/EX(1000),LY(1000)
C
DO 130 J=1,N20
KX=KX+1
EX(KX)=XP(J)
CONTINUE
DO 200 J=1,NR
KY=KY+1
EY(KY)=Y(J)
CONTINUE
RETURN
END
C
SUBROUTINE AVGSIM(XA,Y,N,NR)
DIMENSION XA(10),Y(20),YY(20)
COMMON/BL15/KOUPK,DOUPK
C
I=1 (KOUPK.NE.1) WRITE (12,1120) I
IF (KOUPK.NE.1) WRITE (12,1120) I
CALL SIMUL(XA,Y,NP,1)
RETURN
C
IF (KOUPK.EQ.1) GO TO 1000
DO 300 I=2,KOUPK

```

```

WRITE (12,1120) I
CALL SIMUL(XA,YY,NR,I)
REX=10
DO 200 J=1,NR
  Y(J)=Y(J)+YY(J)
200 CONTINUE
300 CONTINUE
  JU=100 J=1,NR
  Y(J)=Y(J)/JUPK
400 CONTINUE
  SCALE (12,1400)
1000 CALL WRITE(XA,Y,N,NR,1)
C
1120 FORMAT (//IX,'** SIMUL: **',13//)
1400 FORMAT (//IX,'***** AVERAGE RESULT *****//')
END
C
SUBROUTINE SIMUL(XS,Y,IPI,KTER)
  DIMENSION XS(10),Y(20)
  COMMON /SUMMARY/ I,KACT,TKDET,IITSNK,IITCUT,INEUAT,TNESUC,ITNEAL,
  1 FLOST,TWAST,TRUN
  1 COMMON /IPRINTX/ IPRINT(25)
  1 JULINH/IJPOS,UREG
  2 /BLCKH/ENTIM(500),BTIME(300),CNTIM(300)
  3 /BLCKJ/MINTYP(300),MINTC(300)
  4 /BLCKM/NSHTYP(500),ITIME,NHUNT,NSHIP,NIRAFF,JGROUP,NMINE
  5 /BLCKC/DEFT(10),DEPTH(300),TOP(10),DMGSM,THRSW
  1 COMMON/BLCKI/RESTIM(5,10),STIM(5,10),ICIRAN(10),KCURR(50),
  2 /BLCKJ/PVINA(300),PMINY(300)
  3 /BLCKXX/TEMPER,CASIV,FIMJAV,IIMAX
  1 COMMON/BLCKAW/QUENTIM(500),NORMTH,CW2,CDXMAX(45),NAVMAX,NGPI,
  2 /BLCKZZ/MINLCK(300)
  1 COMMON/SAVE/PTORL(100,4),OTURE(100,4),TIIN,POORE(100,4),QPORE(100,
  14)
  1 COMMON/DA4AGE/SOMG(5,10),NFUN,KS*PLI(10),FREQ(16,4),PTUTA(4),OTITA
  1 (4)
  1 COMMON/FLDGG/JFLGG
  1 COMMON/BLDEE4/ITER
  1 COMMON/BLDEE1/XLP(10),XUP(10)
  1 COMMON/BLDEE2/ISCE01,ISEE02
  1 COMMON/BLDEE3/XL(10),XL(10),YL,YU
  1 COMMON/BLDEE16/LSEED
  REAL*8 SEED
C
TKACT=0
TKDET=0
TITSNK=0
TITCUT=0
TNEUAT=0
TNESUC=0
FLOST=0
TWAST=0
C INITIALIZING FOR TOTAL AVERAGES IN OUTPUT
C

```

APPENDIX D

PROGRAM LISTING

FOR A

SECOND-ORDER RESPONSE SURFACE METHOD

BASED ON BOX'S COMPLEX SEARCH


```

C      KY=0
C      DO 100 J=1,N
C        DA(J)=XLP(J)-XLP(J)
C        SECT=JSECT(J)
C        DAP(J)=DA(J)/SECT
C      100 CONTINUE
C
C      IREGN=1
C      DO 300 J=1,N
C        IREGN=IREGN+JSECT(J)-
C      300 CONTINUE
C
C      DO 500 K=1,IREGN
C        KEYS(K)=0
C      500 CONTINUE
C
C      AN=N
C      NP=N+NPBC
C      NPHI=NP-1
C      ANP=NP
C      ANPAI=NPMI
C      NZO=N-(N+3)/2
C
C      AEXP=0
C      ITER=0
C      ITER2=10000
C      IBOX=1
C      CALL BOXCOM(IBOX,IX,KX,KY,XLP,XUP,DA,YIMAX)
C
C      IF(MLEXP.EQ.0) GO TO 1000
C      WRITE (12,1400)
C      CALL COMPEN(KX,KY,ITFY,XLK,XUR,YIMAX)
C
C      IF (ITRY.EQ.0) GO TO 1000
C      ITER2=30000
C      MLEXP=MEXP+MAEXP
C      WRITE (12,5001)
C      CALL BOXCOM(IBOX,IX,KX,KY,XLR,XUR,DAP,YIMAX)
C
C      1000 WRITE (12,5001)
C          CALL REGRFS
C
C      WRITE (12,5001)
C      IBOX=2
C      ITER2=20000
C      CALL BOXCOM(1BOB,IX,KX,KY,XLP,XUP,DA,YIMAX)
C
C      1050 FORMAT (//45X,'***** LIST OF INPUT DATA *****')
C      1400 FORMAT (1H1//45X,'***** COMPENSATE EXPERIMENTS *****')
C      5001 FORMAT (1H1)
C          STOP
C          EN
C
C      SUBROUTINE BOXCOM(1BOB,IX,KX,KY,XL,XU,DA,YIMAX)
C      DIMENSION X(10,30),AP(30),C(10),Y(10),YP(10,10),XL(10),XU(10),DA(10000),
C      & YI(10)

```

```

COMMON N,AN,NM1,N2D,P1,NR,M,MEXP,NP,ANP,NPM1,ANPM1
C*****BLEE1/XLP(10),XUP(10)
C*****BLEE2/ISEED1,ISEED2
C*****BLEE3/ICNK
C*****BLEE4/ITER
C*****BLEE5/MAEXP,SMALL(2)
C*****BLEE6/ITER2
C
C      WRITE (12,1700)
C      MAX=1
C
C      DO 220 I=1,NP
C      DO 140 J=1,N
C      CALL RANDJ(IX,IY,RNUM)
C      IX=IY
C      XP(J)=XL(J)+RNUM*DA(J)
C      XL(J)=XP(J)
C      CONTINUE
C      CALL SECIZE(XP)
C      IF (IBOX.EQ.1) CALL FSBPT1(XP,Y,ICUT)
C      IF (IBOX.EQ.2) CALL FSBPT2(XP,Y,ICUT)
C      IF (ICUT.EQ.2) GO TO 120
C      ITER2=ITER2+1
C      WRITE (12,1450) ITER2
C      CALL WRITE(XP,Y,N,NR,ICUT)
C      IF (IBOX.EQ.2) GO TO 160
C      CALL RECCRD(XP,Y,KX,KY)
C      MEXP=MEXP+1
C      CALL REGION(XP)
C      IF (ICUT.EQ.1) GO TO 120
C      DO 200 J=1,NR
C      YP(J)=Y(J)
C      CONTINUE
C      Y1(1)=Y(1)
C      IF (Y(1).GT.Y1(MAX)) MAX=1
C      CONTINUE
C      WRITE (12,1230) NP
C
C      CALL WORST(Y1,MIN)
C      DO 340 J=1,N
C      SUM=-X(MIN,J)
C      DO 320 I=1,NP
C      SUM=SUM+X(I,J)
C      CONTINUE
C      C(J)=SUM/ANPM1
C      CONTINUE
C
C      R=2**0.5
C      DO 420 J=1,N
C      XP(J)=C(J)+R*(C(J)-X(MIN,J))
C      CONTINUE
C      CALL SECIZE(XP)
C      IF (IBOX.EQ.1) CALL FSBPT1(XP,Y,ICUT)
C      IF (IBOX.EQ.2) CALL FSBPT2(XP,Y,ICUT)
C      IF (ICUT.EQ.2) GO TO 400
C      ITER2=ITER2+1
C      WRITE (12,1450) ITER2

```

```

00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720
00001730
00001740
00001750
00001760
00001770

```

```

00001780
00001790
00001800
00001810
00001820
00001830
00001840
00001850
00001860
00001870
00001880
00001890
00001900
00001910
00001920
00001930
00001940
00001950
00001960
00001970
00001980
00001990
0002000
0002010
0002020
0002030
0002040
0002050
0002060
0002070
0002080
0002090
0002100
0002110
0002120
0002130
0002140
0002150
0002160
0002170
0002180
0002190
0002200
0002210
0002220
0002230
0002240
0002250
0002260
0002270
0002280
0002290
0002300
0002310
0002320
0002330
0002340
0002350
0002360

CALL WRITE(XP,Y,N,NR,ICUT)
CALL RECORD(XP,Y,KX,KY)
WEXP=WEXP+1
CALL REGION(XP)
IF (IEXP.GE.MAEXP.AND.ICUT.EQ.1) GO TO 600
IF (ICUT.EQ.1) GO TO 400
IF (Y(1).LE.Y1(MIN)) GO TO 400
DO 500 J=1,N
  X(MIN,J)=XP(J)
CONTINUE
DO 400 J=1,NR
  YP(MIN,J)=Y(J)
CONTINUE
Y1(MIN)=Y(1)
IF (Y1(MIN).GT.Y1(MAX)) MAX=MIN
IF (IBOX.EQ.1.AND.WEXP.GE.MAEXP) GO TO 600

C
DO 510 I=1,NP
  A=Y1(MAX)-Y1(1)
  IF (A.GT.SMALL(1BX)) GO TO 300
510 CONTINUE

C
IF (ICBK.EQ.0) GO TO 600
DO 520 J=1,N
  SUM=X(1,J)
DO 520 I=2,NP
  SUM=SUM+X(I,J)
CONTINUE
XPC(J)=SUM/ANP
520 CONTINUE

C
CALL SECIZE(XP)
IF (IBOX.FO.1) CALL FSBPT1(XP,Y,ICUT)
IF (IBOX.FO.2) CALL FSBPT2(XP,Y,ICUT)
IF (IBOX.EQ.2) GO TO 540
CALL RECORD(XP,Y,KX,KY)
WEXP=WEXP+1
CALL REGION(XP)
ITER2=ITER2+1
WRITE (12,1950) ITER2
CALL WRITE(XP,Y,N,NR,ICUT)

C
DO 550 I=1,NP
  A=Y(1)-Y1(1)
  IF (A.GT.SMALL(1BOX)) GO TO 560
CONTINUE
GO TO 600

C
560 CALL WORST(Y1,MIN)
IF (Y(1).LE.Y1(MIN)) GO TO 600
DO 570 J=1,N
  X(1,J)=XP(J)
CONTINUE
DO 580 J=1,NR
  YP(MIN,J)=Y(J)
CONTINUE
Y1(MIN)=Y(1)
IF (Y1(MAX).GT.Y1(MIN)) GO TO 300
580

```

```

C 600 MAX=MIN
GO TO 300
WRITE (12,1600)
DO 620 J=1,N
WRITE (12,1160) J,X(MAX,J)
620 CONTINUE
DO 640 J=1,NR
WRITE (12,1180) J,YP(MAX,J)
640 CONTINUE
C
YIMAX=YI(MAX)
IF (180X.EQ.1) RETURN
DO 700 J=1,N
XP(J)=XIMAX,J)
700 CONTINUE
CALL FSOPTI(XP,Y,IOUT)
WRITE (12,1600)
CALL WRITE(XP,Y,N,NR,IOUT)
C
1160 FORMAT (3X,'X(',I1,') = ',F14.6)
1180 FORMAT (3X,'Y(',I1,') = ',F14.6)
1230 FORMAT (//5X,'THE ABOVE ',I2,' POINTS ARE THE INITIAL POINTS'//)
1450 FORMAT (///1X,15,'ITERATION'//)
1500 FORMAT (7X,'** CENTRAL SOLUTION'//)
1600 FORMAT (///15X,'OPTIMAL SOLUTION'//)
1700 FORMAT (//27X,'*****',15X,
E 'INFEASIBLE POINT *****')
C 5000 FORMAT (//)
RETURN
END
C
SUBROUTINE SECIZE(XS)
C ***** THIS SUBROUTINE TRANSFORMS 1ST ORDER INDEPENDENT VARIABLES
C ***** INTO 2ND ORDER INDEPENDENT VARIABLES FOR
C ***** 2ND ORDER REGRESSION
C
DIMENSION XS(30)
COMMON N,AN,NM1,N20,N2DP1,NR,M,MEXP,NP,ANP,NPM1,ANPM1
K=N
DO 100 J=1,N
K=K+1
XS(K)=XS(J)*XS(J)
100 CONTINUE
C
DO 300 I=1,NM1
IP1=I+1
DO 200 J=IP1,N
K=K+1
XS(K)=XS(I)*XS(J)
200 CONTINUE
300 CONTINUE
RETURN
END
C
SUBROUTINE REGRES

```

```

00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850
00002860
00002870
00002880
00002890
00002900
00002910
00002920
00002930
00002940
00002950

```



```

C ***** THIS SUBROUTINE CALCULATES THE REGRESSION COEFFICIENTS OF THE
C ***** RESPONSE SURFACES
C
      DIMENSION XBAR(30),SIO(30),RX(900),RT(465),BV(30),DV(30),IV(30),
      & ISAVE(30),RY(30),SB(30),ANS(10),XE(100),YE(1000)
      COMMON N,AN,NM1,N2D,N2DPI,NR,M,MEXP,NP,ANP1,NPML,ANPML
      COMMON/BL6/B(10,30)
      COMMON/BL7/EX(1000),EY(1000)
      NXE=4EXP#N2D
      N2DPI=N2D+1
C
      WRITE (12,1450)
      DO 600 I=1,MEXP
      KX1=(I-1)*N2D+1
      KX2=I*N2D
      KY1=(I-1)*NR+1
      KY2=I*NR
      WRITE (12,1600) I,(EX(J),J=KX1,KX2),(EY(J),J=KY1,KY2)
      CONTINUE
C
      CALL GMTIRA(LX,XE,N2D,MEXP)
      CALL GMTIRA(EY,YE,NR,MEXP)
C
      DO 50 J=1,N2DPI
      ISAVE(J)=J
      CONTINUE
C
      WRITE (12,1150)
      DO 300 J=1,NR
      KXE=KYE
      KYE=4EXP*(J-1)
      DO 100 I=1,MEXP
      KXE=KXE+1
      KYE=KYE+1
      XF(KXE)=YE(KYE)
      CONTINUE
C
      CALL CORPE(MEXP,N2DPI,1,XL,XBAR,STD,RX,RT,BV,DV,TV)
      CALL CORPE(N2DPI,RT,N2DPI,N2D,ISAVE,RX,RY)
      CALL MINV(RX,N2D,CEI,DV,TV)
      CALL MULTR(MEXP,N2D,XBAR,STD,BV,RX,RY,ISAVE,DV,SB,TV,ANS)
      WRITE (12,1160) (DV(L),L=1,N2D),ANS(1),J
      EVALU=ANS(10)
      WRITE (12,1170) EVALU
C
      DO 200 JCUEFF=1,N2D
      JC(JCUEFF)=DV(JCUEFF)
      CONTINUE
      JC(J,N2DPI)=ANS(1)
      CONTINUE
C
      1450 FORMAT (//45X,'***** REGRESSION TABLEAU *****',//12X,'X1',12X,
      & 'X2',11X,'X1S',10X,'X2S',10X,'Y1',12X,'Y2',12X,'Y3',
      & 'Y4',12X,'Y5',12X,'Y6',12X,'Y7',12X,'Y8',12X,'Y9',12X,'Y10',12X,
      & 'Y11',12X,'Y12',12X,'Y13',12X,'Y14',12X,'Y15',12X,'Y16',12X,'Y17',12X,
      & 'Y18',12X,'Y19',12X,'Y20',12X,'Y21',12X,'Y22',12X,'Y23',12X,'Y24',12X,
      & 'Y25',12X,'Y26',12X,'Y27',12X,'Y28',12X,'Y29',12X,'Y30',12X,'Y31',12X,
      & 'Y32',12X,'Y33',12X,'Y34',12X,'Y35',12X,'Y36',12X,'Y37',12X,'Y38',12X,
      & 'Y39',12X,'Y40',12X,'Y41',12X,'Y42',12X,'Y43',12X,'Y44',12X,'Y45',12X,
      & 'Y46',12X,'Y47',12X,'Y48',12X,'Y49',12X,'Y50',12X,'Y51',12X,'Y52',12X,
      & 'Y53',12X,'Y54',12X,'Y55',12X,'Y56',12X,'Y57',12X,'Y58',12X,'Y59',12X,
      & 'Y60',12X,'Y61',12X,'Y62',12X,'Y63',12X,'Y64',12X,'Y65',12X,'Y66',12X,
      & 'Y67',12X,'Y68',12X,'Y69',12X,'Y70',12X,'Y71',12X,'Y72',12X,'Y73',12X,
      & 'Y74',12X,'Y75',12X,'Y76',12X,'Y77',12X,'Y78',12X,'Y79',12X,'Y80',12X,
      & 'Y81',12X,'Y82',12X,'Y83',12X,'Y84',12X,'Y85',12X,'Y86',12X,'Y87',12X,
      & 'Y88',12X,'Y89',12X,'Y90',12X,'Y91',12X,'Y92',12X,'Y93',12X,'Y94',12X,
      & 'Y95',12X,'Y96',12X,'Y97',12X,'Y98',12X,'Y99',12X,'Y100',12X,'Y101',12X,
      & 'Y102',12X,'Y103',12X,'Y104',12X,'Y105',12X,'Y106',12X,'Y107',12X,'Y108',12X,
      & 'Y109',12X,'Y110',12X,'Y111',12X,'Y112',12X,'Y113',12X,'Y114',12X,'Y115',12X,
      & 'Y116',12X,'Y117',12X,'Y118',12X,'Y119',12X,'Y120',12X,'Y121',12X,'Y122',12X,
      & 'Y123',12X,'Y124',12X,'Y125',12X,'Y126',12X,'Y127',12X,'Y128',12X,'Y129',12X,
      & 'Y130',12X,'Y131',12X,'Y132',12X,'Y133',12X,'Y134',12X,'Y135',12X,'Y136',12X,
      & 'Y137',12X,'Y138',12X,'Y139',12X,'Y140',12X,'Y141',12X,'Y142',12X,'Y143',12X,
      & 'Y144',12X,'Y145',12X,'Y146',12X,'Y147',12X,'Y148',12X,'Y149',12X,'Y150',12X,
      & 'Y151',12X,'Y152',12X,'Y153',12X,'Y154',12X,'Y155',12X,'Y156',12X,'Y157',12X,
      & 'Y158',12X,'Y159',12X,'Y160',12X,'Y161',12X,'Y162',12X,'Y163',12X,'Y164',12X,
      & 'Y165',12X,'Y166',12X,'Y167',12X,'Y168',12X,'Y169',12X,'Y170',12X,'Y171',12X,
      & 'Y172',12X,'Y173',12X,'Y174',12X,'Y175',12X,'Y176',12X,'Y177',12X,'Y178',12X,
      & 'Y179',12X,'Y180',12X,'Y181',12X,'Y182',12X,'Y183',12X,'Y184',12X,'Y185',12X,
      & 'Y186',12X,'Y187',12X,'Y188',12X,'Y189',12X,'Y190',12X,'Y191',12X,'Y192',12X,
      & 'Y193',12X,'Y194',12X,'Y195',12X,'Y196',12X,'Y197',12X,'Y198',12X,'Y199',12X,
      & 'Y200',12X,'Y201',12X,'Y202',12X,'Y203',12X,'Y204',12X,'Y205',12X,'Y206',12X,
      & 'Y207',12X,'Y208',12X,'Y209',12X,'Y210',12X,'Y211',12X,'Y212',12X,'Y213',12X,
      & 'Y214',12X,'Y215',12X,'Y216',12X,'Y217',12X,'Y218',12X,'Y219',12X,'Y220',12X,
      & 'Y221',12X,'Y222',12X,'Y223',12X,'Y224',12X,'Y225',12X,'Y226',12X,'Y227',12X,
      & 'Y228',12X,'Y229',12X,'Y230',12X,'Y231',12X,'Y232',12X,'Y233',12X,'Y234',12X,
      & 'Y235',12X,'Y236',12X,'Y237',12X,'Y238',12X,'Y239',12X,'Y240',12X,'Y241',12X,
      & 'Y242',12X,'Y243',12X,'Y244',12X,'Y245',12X,'Y246',12X,'Y247',12X,'Y248',12X,
      & 'Y249',12X,'Y250',12X,'Y251',12X,'Y252',12X,'Y253',12X,'Y254',12X,'Y255',12X,
      & 'Y256',12X,'Y257',12X,'Y258',12X,'Y259',12X,'Y260',12X,'Y261',12X,'Y262',12X,
      & 'Y263',12X,'Y264',12X,'Y265',12X,'Y266',12X,'Y267',12X,'Y268',12X,'Y269',12X,
      & 'Y270',12X,'Y271',12X,'Y272',12X,'Y273',12X,'Y274',12X,'Y275',12X,'Y276',12X,
      & 'Y277',12X,'Y278',12X,'Y279',12X,'Y280',12X,'Y281',12X,'Y282',12X,'Y283',12X,
      & 'Y284',12X,'Y285',12X,'Y286',12X,'Y287',12X,'Y288',12X,'Y289',12X,'Y290',12X,
      & 'Y291',12X,'Y292',12X,'Y293',12X,'Y294',12X,'Y295',12X,'Y296',12X,'Y297',12X,
      & 'Y298',12X,'Y299',12X,'Y300',12X,'Y301',12X,'Y302',12X,'Y303',12X,'Y304',12X,
      & 'Y305',12X,'Y306',12X,'Y307',12X,'Y308',12X,'Y309',12X,'Y310',12X,'Y311',12X,
      & 'Y312',12X,'Y313',12X,'Y314',12X,'Y315',12X,'Y316',12X,'Y317',12X,'Y318',12X,
      & 'Y319',12X,'Y320',12X,'Y321',12X,'Y322',12X,'Y323',12X,'Y324',12X,'Y325',12X,
      & 'Y326',12X,'Y327',12X,'Y328',12X,'Y329',12X,'Y330',12X,'Y331',12X,'Y332',12X,
      & 'Y333',12X,'Y334',12X,'Y335',12X,'Y336',12X,'Y337',12X,'Y338',12X,'Y339',12X,
      & 'Y340',12X,'Y341',12X,'Y342',12X,'Y343',12X,'Y344',12X,'Y345',12X,'Y346',12X,
      & 'Y347',12X,'Y348',12X,'Y349',12X,'Y350',12X,'Y351',12X,'Y352',12X,'Y353',12X,
      & 'Y354',12X,'Y355',12X,'Y356',12X,'Y357',12X,'Y358',12X,'Y359',12X,'Y360',12X,
      & 'Y361',12X,'Y362',12X,'Y363',12X,'Y364',12X,'Y365',12X,'Y366',12X,'Y367',12X,
      & 'Y368',12X,'Y369',12X,'Y370',12X,'Y371',12X,'Y372',12X,'Y373',12X,'Y374',12X,
      & 'Y375',12X,'Y376',12X,'Y377',12X,'Y378',12X,'Y379',12X,'Y380',12X,'Y381',12X,
      & 'Y382',12X,'Y383',12X,'Y384',12X,'Y385',12X,'Y386',12X,'Y387',12X,'Y388',12X,
      & 'Y389',12X,'Y390',12X,'Y391',12X,'Y392',12X,'Y393',12X,'Y394',12X,'Y395',12X,
      & 'Y396',12X,'Y397',12X,'Y398',12X,'Y399',12X,'Y400',12X,'Y401',12X,'Y402',12X,
      & 'Y403',12X,'Y404',12X,'Y405',12X,'Y406',12X,'Y407',12X,'Y408',12X,'Y409',12X,
      & 'Y410',12X,'Y411',12X,'Y412',12X,'Y413',12X,'Y414',12X,'Y415',12X,'Y416',12X,
      & 'Y417',12X,'Y418',12X,'Y419',12X,'Y420',12X,'Y421',12X,'Y422',12X,'Y423',12X,
      & 'Y424',12X,'Y425',12X,'Y426',12X,'Y427',12X,'Y428',12X,'Y429',12X,'Y430',12X,
      & 'Y431',12X,'Y432',12X,'Y433',12X,'Y434',12X,'Y435',12X,'Y436',12X,'Y437',12X,
      & 'Y438',12X,'Y439',12X,'Y440',12X,'Y441',12X,'Y442',12X,'Y443',12X,'Y444',12X,
      & 'Y445',12X,'Y446',12X,'Y447',12X,'Y448',12X,'Y449',12X,'Y450',12X,'Y451',12X,
      & 'Y452',12X,'Y453',12X,'Y454',12X,'Y455',12X,'Y456',12X,'Y457',12X,'Y458',12X,
      & 'Y459',12X,'Y460',12X,'Y461',12X,'Y462',12X,'Y463',12X,'Y464',12X,'Y465',12X,
      & 'Y466',12X,'Y467',12X,'Y468',12X,'Y469',12X,'Y470',12X,'Y471',12X,'Y472',12X,
      & 'Y473',12X,'Y474',12X,'Y475',12X,'Y476',12X,'Y477',12X,'Y478',12X,'Y479',12X,
      & 'Y480',12X,'Y481',12X,'Y482',12X,'Y483',12X,'Y484',12X,'Y485',12X,'Y486',12X,
      & 'Y487',12X,'Y488',12X,'Y489',12X,'Y490',12X,'Y491',12X,'Y492',12X,'Y493',12X,
      & 'Y494',12X,'Y495',12X,'Y496',12X,'Y497',12X,'Y498',12X,'Y499',12X,'Y500',12X,
      & 'Y501',12X,'Y502',12X,'Y503',12X,'Y504',12X,'Y505',12X,'Y506',12X,'Y507',12X,
      & 'Y508',12X,'Y509',12X,'Y510',12X,'Y511',12X,'Y512',12X,'Y513',12X,'Y514',12X,
      & 'Y515',12X,'Y516',12X,'Y517',12X,'Y518',12X,'Y519',12X,'Y520',12X,'Y521',12X,
      & 'Y522',12X,'Y523',12X,'Y524',12X,'Y525',12X,'Y526',12X,'Y527',12X,'Y528',12X,
      & 'Y529',12X,'Y530',12X,'Y531',12X,'Y532',12X,'Y533',12X,'Y534',12X,'Y535',12X,
      & 'Y536',12X,'Y537',12X,'Y538',12X,'Y539',12X,'Y540',12X,'Y541',12X,'Y542',12X,
      & 'Y543',12X,'Y544',12X,'Y545',12X,'Y546',12X,'Y547',12X,'Y548',12X,'Y549',12X,
      & 'Y550',12X,'Y551',12X,'Y552',12X,'Y553',12X,'Y554',12X,'Y555',12X,'Y556',12X,
      & 'Y557',12X,'Y558',12X,'Y559',12X,'Y560',12X,'Y561',12X,'Y562',12X,'Y563',12X,
      & 'Y564',12X,'Y565',12X,'Y566',12X,'Y567',12X,'Y568',12X,'Y569',12X,'Y570',12X,
      & 'Y571',12X,'Y572',12X,'Y573',12X,'Y574',12X,'Y575',12X,'Y576',12X,'Y577',12X,
      & 'Y578',12X,'Y579',12X,'Y580',12X,'Y581',12X,'Y582',12X,'Y583',12X,'Y584',12X,
      & 'Y585',12X,'Y586',12X,'Y587',12X,'Y588',12X,'Y589',12X,'Y590',12X,'Y591',12X,
      & 'Y592',12X,'Y593',12X,'Y594',12X,'Y595',12X,'Y596',12X,'Y597',12X,'Y598',12X,
      & 'Y599',12X,'Y600',12X,'Y601',12X,'Y602',12X,'Y603',12X,'Y604',12X,'Y605',12X,
      & 'Y606',12X,'Y607',12X,'Y608',12X,'Y609',12X,'Y610',12X,'Y611',12X,'Y612',12X,
      & 'Y613',12X,'Y614',12X,'Y615',12X,'Y616',12X,'Y617',12X,'Y618',12X,'Y619',12X,
      & 'Y620',12X,'Y621',12X,'Y622',12X,'Y623',12X,'Y624',12X,'Y625',12X,'Y626',12X,
      & 'Y627',12X,'Y628',12X,'Y629',12X,'Y630',12X,'Y631',12X,'Y632',12X,'Y633',12X,
      & 'Y634',12X,'Y635',12X,'Y636',12X,'Y637',12X,'Y638',12X,'Y639',12X,'Y640',12X,
      & 'Y641',12X,'Y642',12X,'Y643',12X,'Y644',12X,'Y645',12X,'Y646',12X,'Y647',12X,
      & 'Y648',12X,'Y649',12X,'Y650',12X,'Y651',12X,'Y652',12X,'Y653',12X,'Y654',12X,
      & 'Y655',12X,'Y656',12X,'Y657',12X,'Y658',12X,'Y659',12X,'Y660',12X,'Y661',12X,
      & 'Y662',12X,'Y663',12X,'Y664',12X,'Y665',12X,'Y666',12X,'Y667',12X,'Y668',12X,
      & 'Y669',12X,'Y670',12X,'Y671',12X,'Y672',12X,'Y673',12X,'Y674',12X,'Y675',12X,
      & 'Y676',12X,'Y677',12X,'Y678',12X,'Y679',12X,'Y680',12X,'Y681',12X,'Y682',12X,
      & 'Y683',12X,'Y684',12X,'Y685',12X,'Y686',12X,'Y687',12X,'Y688',12X,'Y689',12X,
      & 'Y690',12X,'Y691',12X,'Y692',12X,'Y693',12X,'Y694',12X,'Y695',12X,'Y696',12X,
      & 'Y697',12X,'Y698',12X,'Y699',12X,'Y700',12X,'Y701',12X,'Y702',12X,'Y703',12X,
      & 'Y704',12X,'Y705',12X,'Y706',12X,'Y707',12X,'Y708',12X,'Y709',12X,'Y710',12X,
      & 'Y711',12X,'Y712',12X,'Y713',12X,'Y714',12X,'Y715',12X,'Y716',12X,'Y717',12X,
      & 'Y718',12X,'Y719',12X,'Y720',12X,'Y721',12X,'Y722',12X,'Y723',12X,'Y724',12X,
      & 'Y725',12X,'Y726',12X,'Y727',12X,'Y728',12X,'Y729',12X,'Y730',12X,'Y731',12X,
      & 'Y732',12X,'Y733',12X,'Y734',12X,'Y735',12X,'Y736',12X,'Y737',12X,'Y738',12X,
      & 'Y739',12X,'Y740',12X,'Y741',12X,'Y742',12X,'Y743',12X,'Y744',12X,'Y745',12X,
      & 'Y746',12X,'Y747',12X,'Y748',12X,'Y749',12X,'Y750',12X,'Y751',12X,'Y752',12X,
      & 'Y753',12X,'Y754',12X,'Y755',12X,'Y756',12X,'Y757',12X,'Y758',12X,'Y759',12X,
      & 'Y760',12X,'Y761',12X,'Y762',12X,'Y763',12X,'Y764',12X,'Y765',12X,'Y766',12X,
      & 'Y767',12X,'Y768',12X,'Y769',12X,'Y770',12X,'Y771',12X,'Y772',12X,'Y773',12X,
      & 'Y774',12X,'Y775',12X,'Y776',12X,'Y777',12X,'Y778',12X,'Y779',12X,'Y780',12X,
      & 'Y781',12X,'Y782',12X,'Y783',12X,'Y784',12X,'Y785',12X,'Y786',12X,'Y787',12X,
      & 'Y788',12X,'Y789',12X,'Y790',12X,'Y791',12X,'Y792',12X,'Y793',12X,'Y794',12X,
      & 'Y795',12X,'Y796',12X,'Y797',12X,'Y798',12X,'Y799',12X,'Y800',12X,'Y801',12X,
      & 'Y802',12X,'Y803',12X,'Y804',12X,'Y805',12X,'Y806',12X,'Y807',12X,'Y808',12X,
      & 'Y809',12X,'Y810',12X,'Y811',12X,'Y812',12X,'Y813',12X,'Y814',12X,'Y815',12X,
      & 'Y816',12X,'Y817',12X,'Y818',12X,'Y819',12X,'Y820',12X,'Y821',12X,'Y822',12X,
      & 'Y823',12X,'Y824',12X,'Y825',12X,'Y826',12X,'Y827',12X,'Y828',12X,'Y829',12X,
      & 'Y830',12X,'Y831',12X,'Y832',12X,'Y833',12X,'Y834',12X,'Y835',12X,'Y836',12X,
      & 'Y837',12X,'Y838',12X,'Y839',12X,'Y840',12X,'Y841',12X,'Y842',12X,'Y843',12X,
      & 'Y844',12X,'Y845',12X,'Y846',12X,'Y847',12X,'Y848',12X,'Y849',12X,'Y850',12X,
      & 'Y851',12X,'Y852',12X,'Y853',12X,'Y854',12X,'Y855',12X,'Y856',12X,'Y857',12X,
      & 'Y858',12X,'Y859',12X,'Y860',12X,'Y861',12X,'Y862',12X,'Y863',12X,'Y864',12X,
      & 'Y865',12X,'Y866',12X,'Y867',12X,'Y868',12X,'Y869',12X,'Y870',12X,'Y871',12X,
      & 'Y872',12X,'Y873',12X,'Y874',12X,'Y875',12X,'Y876',12X,'Y877',12X,'Y878',12X,
      & 'Y879',12X,'Y880',12X,'Y881',12X,'Y882',12X,'Y883',12X,'Y884',12X,'Y885',12X,
      & 'Y886',12X,'Y887',12X,'Y888',12X,'Y889',12X,'Y890',12X,'Y891',12X,'Y892',12X,
      & 'Y893',12X,'Y894',12X,'Y895',12X,'Y896',12X,'Y897',12X,'Y898',12X,'Y899',12X,
      & 'Y900',12X,'Y901',12X,'Y902',12X,'Y903',12X,'Y904',12X,'Y905',12X,'Y906',12X,
      & 'Y907',12X,'Y908',12X,'Y909',12X,'Y910',12X,'Y911',12X,'Y912',12X,'Y913',12X,
      & 'Y914',12X,'Y915',12X,'Y916',12X,'Y917',12X,'Y918',12X,'Y919',12X,'Y920',12X,
      & 'Y921',12X,'Y922',12X,'Y923',12X,'Y924',12X,'Y925',12X,'Y926',12X,'Y927',12X,
      & 'Y928',12X,'Y929',12X,'Y930',12X,'Y931',12X,'Y932',12X,'Y933',12X,'Y934',12X,
      & 'Y935',12X,'Y936',12X,'Y937',12X,'Y938',12X,'Y939',12X,'Y940',12X,'Y941',12X,
      & 'Y942',12X,'Y943',12X,'Y944',12X,'Y945',12X,'Y946',12X,'Y947',12X,'Y948',12X,
      & 'Y949',12X,'Y950',12X,'Y951',12X,'Y952',12X,'Y953',12X,'Y954',12X,'Y955',12X,
      & 'Y956',12X,'Y957',12X,'Y958',12X,'Y959',12X,'Y960',12X,'Y961',12X,'Y962',12X,
      & 'Y963',12X,'Y964',12X,'Y965',12X,'Y966',12X,'Y967',12X,'Y968',12X,'Y969',12X,
      & 'Y970',12X,'Y971',12X,'Y972',12X,'Y973',12X,'Y974',12X,'Y975',12X,'Y976',12X,
      & 'Y977',12X,'Y978',12X,'Y979',12X,'Y980',12X,'Y981',12X,'Y982',12X,'Y983',12X,
      & 'Y984',12X,'Y985',12X,'Y986',12X,'Y987',12X,'Y988',12X,'Y989',12X,'Y990',12X,
      & 'Y991',12X,'Y992',12X,'Y993',12X,'Y994',12X,'Y995',12X,'Y996',12X,'Y997',12X,
      & 'Y998',12X,'Y999',12X,'Y1000',12X,'Y1001',12X,'Y1002',12X,'Y1003',12X,'Y1004',12X,
      & 'Y1005',12X,'Y1006',12X,'Y1007',12X,'Y1008',12X,'Y1009',12X,'Y1010',12X,'Y1011',12X,
      & 'Y1012',12X,'Y1013',12X,'Y1014',12X,'Y1015',12X,'Y1016',12X,'Y1017',12X,'Y1018',12X,
      & 'Y1019',12X,'Y1020',12X,'Y1021',12X,'Y1022',12X,'Y1023',12X,'Y1024',12X,'Y1025',12X,
      & 'Y1026',12X,'Y1027',12X,'Y1028',12X,'Y1029',12X,'Y1030',12X,'Y1031',12X,'Y1032',12X,
      & 'Y1033',12X,'Y1034',12X,'Y1035',12X,'Y1036',12X,'Y1037',12X,'Y1038',12X,'Y1039',12X,
      & 'Y1040',12X,'Y1041',12X,'Y1042',12X,'Y1043',12X,'Y1044',12X,'Y1045',12X,'Y1046',12X,
      & 'Y1047',12X,'Y1048',12X,'Y1049',12X,'Y1050',12X,'Y1051',12X,'Y1052',12X,'Y1053',12X,
      & 'Y1054',12X,'Y1055',12X,'Y1056',12X,'Y1057',12X,'Y1058',12X,'Y1059',12X,'Y1060',12X,
      & 'Y1061',12X,'Y1062',12X,'Y1063',12X,'Y1064',12X,'Y1065',12X,'Y1066',12X,'Y1067',12X,
      & 'Y1068',12X,'Y1069',12X,'Y1070',12X,'Y1071',12X,'Y1072',12X,'Y1073',12X,'Y1074',12X,
      & 'Y1075',12X,'Y1076',12X,'Y1077',12X,'Y1078',12X,'Y1079',12X,'Y1080',12X,'Y1081',12X,
      & 'Y1082',12X,'Y1083',12X,'Y1084',12X,'Y1085',12X,'Y1086',12X,'Y1087',12X,'Y1088',12X,
      & 'Y1089',12X,'Y1090',12X,'Y1091',12X,'Y1092',12X,'Y1093',12X,'Y1094',12X,'Y1095',12X,
      & 'Y1096',12X,'Y1097',12X,'Y1098',12X,'Y1099',12X,'Y1100',12X,'Y1101',12X,'Y1102',12X,
      & 'Y1103',12X,'Y1104',12X,'Y1105',12X,'Y1106',12X,'Y1107',12X,'Y1108',12X,'Y1109',12X,
      & 'Y1110',12X,'Y1111',12X,'Y1112',12X,'Y1113',12X,'Y1114',12X,'Y1115',12X,'Y1116',12X,
      & 'Y1117',12X,'Y1118',12X,'Y1119',12X,'Y1120',12X,'Y1121',12X,'Y1122',12X,'Y1123',12X,
      & 'Y1124',12X,'Y1125',12X,'Y1126',12X,'Y1127',12X,'Y1128',12X,'Y1129',12X,'Y1130',12X,
      & 'Y1131',12X,'Y1132',12X,'Y1133',12X,'Y1134',12X,'Y1135',12X,'Y1136',12X,'Y1137',12X,
      & 'Y1138',12X,'Y1139',12X,'Y1140',12X,'Y1141',12X,'Y1142',12X,'Y1143',12X,'Y1144',12X,
      & 'Y1145',12X,'Y1146',12X,'Y1147',12X,'Y1148',12X,'Y1149',12X,'Y1150',12X,'Y1151',12X,
      & 'Y1152',12X,'Y1153',12X,'Y1154',12X,'Y1155',12X,'Y1156',12X,'Y1157',12X,'Y1158',12X,
      & 'Y1159',12X,'Y1160',12X,'Y1161',12X,'Y1162',12X,'Y1163',12X,'Y1164',12X,'Y1165',12X,
      & 'Y1166',12X,'Y1167',12X,'Y1168',12X,'Y1169',12X,'Y1170
```

```

1160 FORMAT (/5X,F12.6,' X(1) + ',F12.6,' X(2) + ',F12.6,'
      &' X(1)SQ + ',F12.6,' X(2)SQ + ',F12.6,' X(1)*X(2) + ',
      &' F12.6,' = Y(',F12.6,' IL,',F12.6,' )')
1170 FORMAT (/5X,'F VALUE = ',F12.6)
      RETURN
      END
C
C SUBROUTINE FSBPT1(XT,Y,ICUT)
C
C ***** THIS SUBROUTINE INDICATES IF THE CURRENT POINT IS FEASIBLE
C ***** ICUT=1 : VIOLATES THE IMPLICIT CONSTRAINT(S)
C ***** ICUT=2 : VIOLATES THE EXPLICIT CONSTRAINT(S)
C
      DIMENSION XT(30),Y(10),YY(10)
      COMMON/AN,M1,N2,N2UP1,NR,M,MLXP,NP,ANP,NPML,ANPML
      COMMON/BLUEE1/XLP(10),XUP(10)
      COMMON/BLUEE3/AL(10),AU(10),YL,YU
      COMMON/BL15/KCUPK,DUPK
      ICUT=2
      DO 100 J=1,N
        IF (XT(J).LT.XLP(J)) GO TO 500
        IF (XT(J).GT.XUP(J)) GO TO 500
      100 CONTINUE
      ICUT=1
C
      IF (KCUPK.NE.1) WRITE (12,1120) I
      CALL SIMUL(XT,Y,NR,1)
      RETURN 10
      IF (KCUPK.EQ.1) GO TO 190
      DO 160 I=2,KDUPK
        WRITE (12,1120) I
        CALL SIMUL(XT,Y,NR,2)
        RETURN 10
      DO 140 J=1,NR
        Y(J)=Y(J)+YY(J)
      140 CONTINUE
      160 CONTINUE
      Y(J)=Y(J)/DUPK
      180 CONTINUE
C
      DO 190 J=2,NR
        IF (Y(J).LT.YL) GO TO 500
        IF (Y(J).GT.YU) GO TO 500
      200 CONTINUE
      ICUT=0
C
1120 FORMAT (/1X,'** SIMUL.',I3,' **/')
500 RETURN
      EN
C
C SUBROUTINE FSBPT2(XT,Y,ICUT)
C
C ***** THIS SUBROUTINE CALCULATES THE RESPONSES FROM THE REGRESSION
C ***** FUNCTIONS AND INDICATES IF THE CURRENT POINT IS FEASIBLE
C ***** ICUT=1 : VIOLATES THE IMPLICIT CONSTRAINT(S)
C ***** ICUT=2 : VIOLATES THE EXPLICIT CONSTRAINT(S)
C

```

```

00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060
00004070
00004080
00004090
00004100
00004110
00004120
00004130

```

```

DIMENSION XT(20),Y(10)
COMMON N,AN,NM1,N2D,N2DPL,NR,M,MEXP,NP,ANP,NPM1,ANPM1
COMMON/ALCE1/XL(10),XU(10),YL,YU
COMMON/ALCE1/XLP(10),XUP(10)
COMMON/BL6/B(10,30)
IOJ1=2
DO 100 J=1,N
  IF (XT(J)).LT.XLP(J)) GO TO 500
  IF (XT(J)).GT.XUP(J)) GO TO 500
100 CONTINUE
IOJ1=1
DO 300 I=2,NR
  Y(I)=B(I,N2DPL)
DO 200 J=1,N2D
  Y(I)=Y(I)+B(I,J)*XT(J)
200 CONTINUE
  IF (Y(I)).LT.YL) GO TO 500
  IF (Y(I)).GT.YU) GO TO 500
300 CONTINUE
  Y(I)=B(I,N2DPL)
DO 350 J=1,N2D
  Y(I)=Y(I)+B(I,J)*XT(J)
350 CONTINUE
  IOJ1=0
500 RETURN
END
C
C SUBROUTINE WORST(YI,MIN)
C ***** THIS SUBROUTINE FINDS OUT THE WORST POINT IN THE
C ***** POLYGON
C
DIMENSION YI(30)
COMMON N,AN,NM1,N2D,N2DPL,NR,M,MEXP,NP,ANP,NPM1,ANPM1
MIN=1
YMIN=YI(1)
DO 50 J=2,NP
  IF (YI(J).GE.YMIN) GO TO 50
MIN=J
YMIN=YI(J)
50 CONTINUE
RETURN
END
C
C SUBROUTINE DATA(NV,DV)
C ***** THIS SUBROUTINE IS CALLED BY THE SSP SUBROUTINE 'CORRE'
C ***** TO READ IN THE EXPERIMENT DATA NEEDED FOR REGRESSION
C
DIMENSION DV(1)
FORIAT (12F6.0)
1 READ (3,1) (DV(J),J=1,NV)
WRITE (1) (DV(J),J=1,NV)
RETURN
END

```

```

C
SUBROUTINE WRITE(XA,YA,N,NR,IGUT)
DIMENSION XA(50),YA(10)
C
IF (IGUT.NE.0) GO TO 500
C
DO 100 J=1,N
WRITE (12,1100) J,XA(J)
CONTINUE
100
DO 200 J=1,NR
WRITE (12,1200) J,YA(J)
CONTINUE
GO TO 1000
C
DO 500 J=1,N
WRITE (12,1600) J,XA(J)
CONTINUE
600
DO 700 J=1,NR
WRITE (12,1700) J,YA(J)
CONTINUE
700
FORMAT (30X,'X(',11,') = ',F14.6)
1100
FORMAT (30X,'Y(',11,') = ',F14.6)
1200
FORMAT (80X,'X(',11,') = ',F14.6)
1400
FORMAT (80X,'Y(',11,') = ',F14.6)
1600
FORMAT (80X,'X(',11,') = ',F14.6)
1700
FORMAT (80X,'Y(',11,') = ',F14.6)
1800
RETURN
END
C
SUBROUTINE RECORD(XP,Y,KX,KY)
DIMENSION XP(30),Y(10)
COMMON N,AN,NPM1,N20,N20PL,NR,M,MEXP,NP,ANP,NPM1,ANPM1
COMMON/BL9/EX(1000),LY(1000)
C
DO 100 J=1,N20
KX=KX+1
EX(KX)=XP(J)
CONTINUE
100
DO 200 J=1,NR
KY=KY+1
EY(KY)=Y(J)
CONTINUE
200
RETURN
END
C
SUBROUTINE REGION(XP)
DIMENSION XP(10),JREGN(10)
COMMON N,AN,NPM1,N20,N20PL,NR,M,MEXP,NP,ANP,NPM1,ANPM1
COMMON/BL1/XLP(10),XUP(10)
COMMON/BL2/EE3/XL(10),XU(10),YL,YU
COMMON/BL11/JSECT(10),DAP(10)
COMMON/BL12/KEXRG(1000)
COMMON/BL13/JREGN,MILXR
C
DO 300 J=1,N
KREGN=(XP(J)-XLP(J))/DAP(J)
JREGN(J)=KREGN+1
IF (KREGN.EQ.JSECT(J)) JREGN(J)=JSECT(J)
CONTINUE
300

```

```

00004730
00004740
00004750
00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005080
00005090
00005100
00005110
00005120
00005130
00005140
00005150
00005160
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005310

```

```

K=JREGN(N)
DO 500 JP=1,NM1
J=N-JP
K=(K-1)*JSECT(J)+JREGN(J)
500 CONTINUE
C
KEXRG(K)=KEXRG(K)+1
RETURN
END
C
SUBROUTINE COMPEN(KX,KY,IIRY,XLR,XUR,YIMAX)
DIMENSION XP(10),Y(10),XLR(10),XUR(10),XLRG(10),XURG(10)
COMMON N,AN,NM1,N2D,N2UP1,NR,M,MEXF,NP,ANP,NPM1,ANPM1
COMMON/BL1/XLP(10),XUP(10)
COMMON/BL2/XL(10),XU(10),YL,YU
COMMON/BL11/JSECT(10),DAP(10)
COMMON/BL12/KEXRG(1000)
COMMON/BL13/IREGN,MILX
IIRY=0
ICOMP=0
DO 500 K=1,IREGN
IF(KEXRG(K).GE.MIEXR) GO TO 500
XA=X
DO 300 J=1,N
KB=MDU(KA,JSECT(J))
IF(KB.EQ.0) KB=JSECT(J)
KA=(KA-KB)/JSECT(J)+1
AKR=KB
XLRG(J)=XLP(J)+DAP(J)*(AKR-1.)
XUP(J)=XUP(J)+DAP(J)*AKB
XP(J)=(XLRG(J)+XURG(J))/2.
300 CONTINUE
MEXP=MEXP+1
ICOMP=ICOMP+1
WRITE(12,1400) ICOMP
CALL SECIZE(XP)
CALL FORPI(XP,Y,ICUT)
CALL FORPD(XP,Y,KX,KY)
CALL WRITE(XP,Y,N,NR,ICUT)
IF(1001.EQ.1) GO TO 500
IF(Y(1).LE.YIMAX) GO TO 500
YIMAX=Y(1)
DO 400 J=1,N
XLRG(J)=XLRG(J)
XURG(J)=XURG(J)
400 CONTINUE
IIRY=1
500 CONTINUE
C
1400 FORMAT(1X,I4,' TH COMPENSATE POINT')
RETURN
END
C
SUBROUTINE SIMUL(XS,Y,IIP1,KTER)
DIMENSION XS(10),Y(20)
COMMON /SUMRY/ TRACI,TKDEI,ITISNK,ITICUI,TNEUAT,TNESUC,TNUEAL,
I,ITLOST,ITWAST,IRUN

```


DISTRIBUTION LIST

<u>Name</u>	<u>Number of copies</u>
Office of Naval Research Arlington, VA 22217	
Code 431	2
Code 434	1
Code 436	1
Defense Documentation Center Cameron Station Alexandria, VA 22314	12
ONR Branch Office 536 South Clark Street Chicago, Illinois 60605	1
Naval Research Laboratory Code 2627 Washington, DC 20375	6
ONR Resident Representative Purdue University Room 85 - Graduate House West Lafayette, Indiana 47907	1
Desmatics, Inc. PO Box 618 State College, PA 16801	1
George S. Fishman Curriculum in Operations Research and Systems Analysis Phillips Annex University of North Carolina at Chapel Hill Chapel Hill, NC 27514	1
Donald P. Gaver Naval Postgraduate School Operations Research Department Monterey, CA 93940	1
Toke Jayachandran Naval Postgraduate School Monterey, CA 93940	1
D. L. Iglehart Department of Operations Research Stanford University Stanford, CA 94305	1

AD-A062 389

NOTRE DAME UNIV IND DEPT OF AEROSPACE AND MECHANICA--ETC F/G 12/2
OPTIMIZATION OF MULTIPLE-RESPONSE SIMULATION MODELS.(U)
OCT 78 W E BILES

N00014-76-C-1021
NL

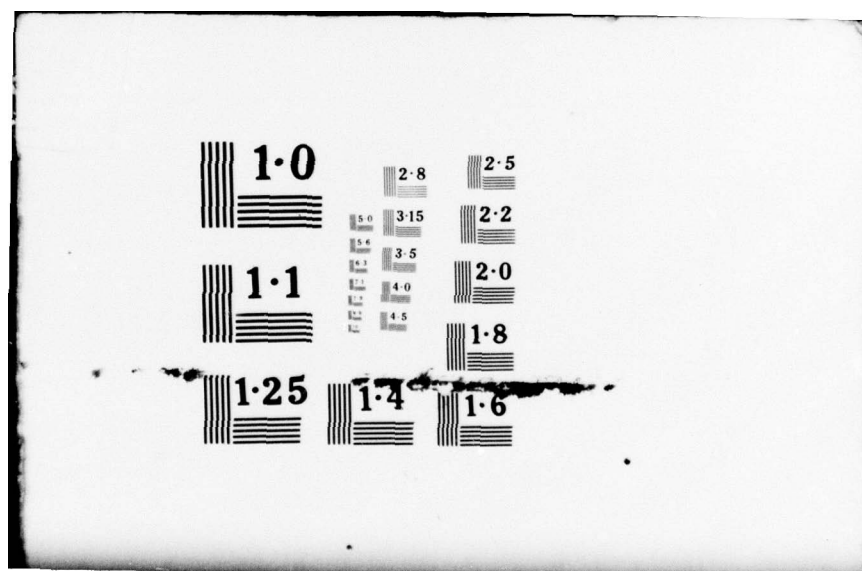
UNCLASSIFIED

2 OF 2
ADIA
082389



END
DATE
FILMED

3 -79
DDC



DISTRIBUTION LIST

<u>Name</u>	<u>Number of copies</u>
Dr. Averill M. Law 1513 University Avenue Department of Industrial Engineering University of Wisconsin Madison, Wisconsin 53706	1
Dr. Peter A. W. Lewis Naval Postgraduate School Department of Administrative Sciences Monterey, CA 93940	1
Dr. Richard E. Nance Computer Science, Head Virginia Polytechnic Institute and State University Blacksburg, VA 24061	1
Dr. A. Alan B. Pritsker Department of Industrial Engineering Purdue University West Lafayette, Indiana 47907	1
J. William Schmidt Department of Industrial Engineering and Operations Research Virginia Polytechnic Institute and State University Blacksburg, VA 24061	1
Naval Surface Weapons Center Code K-30 (Kulp and Patton, and Miller) Dahlgren, VA 22448	3
David W. Taylor Naval Ship Research & Development Center Code 1806 (S. Goldstein) Bethesda, MD 20084	1
Naval Research Laboratory Code 5308 Washington, DC 20375	1
Naval Ocean Systems Center Code 16 San Diego, CA 92152	1